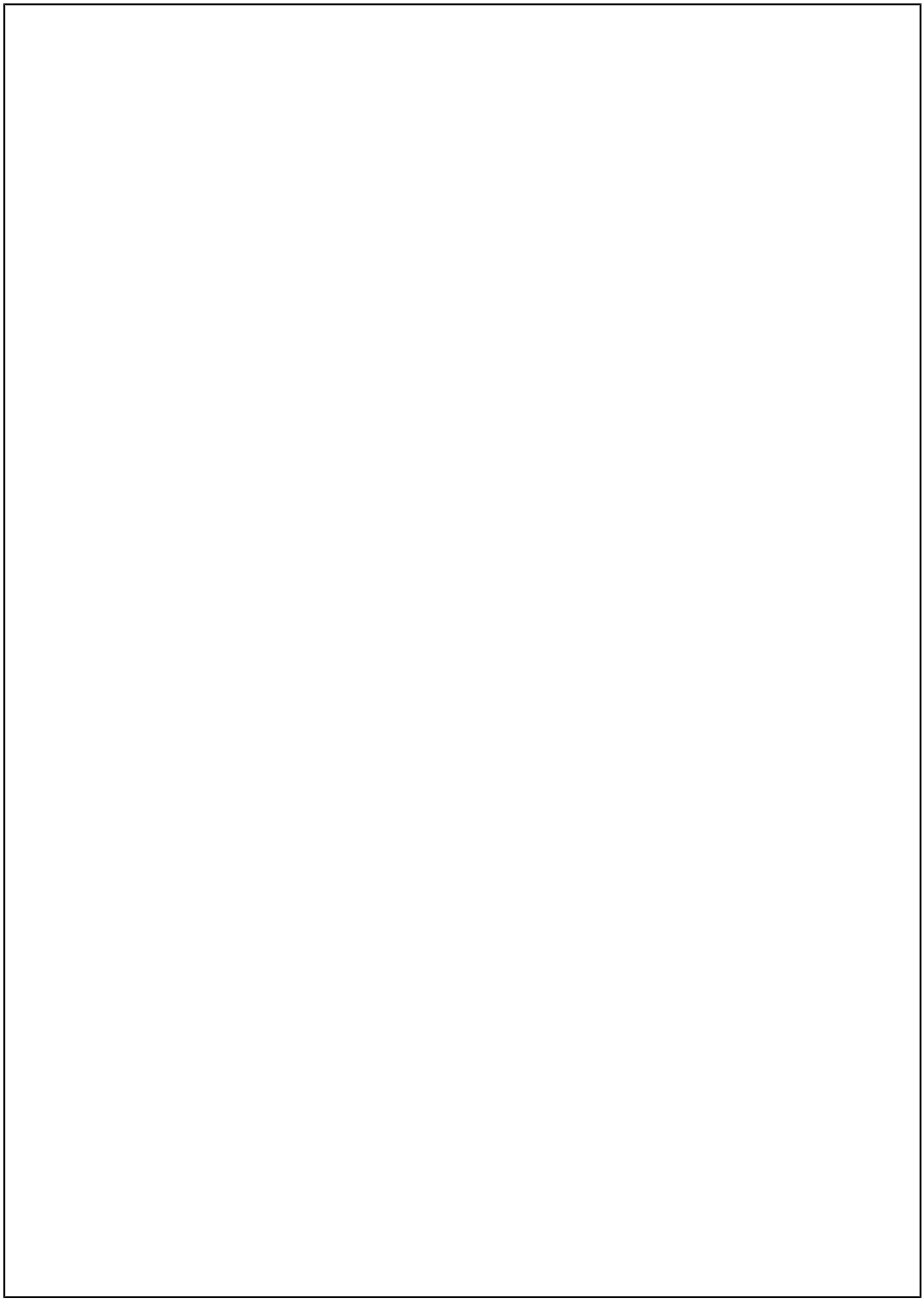
TOSHIBA

32 Bit RISC Microcontroller
TX03 Series

TMPM3U0FSDMG

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION



General precautions on the use of Toshiba MCUs

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

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

1. The MCUs' operation at power-on

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

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

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

2. Unused pins

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

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

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

3. Clock oscillation stability

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

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

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

The SFR 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

注） SAMCR register address is 32 bits wide from the address 0x0000 _ 0004 (Base Address(0x00000000) + unique address (0x0004)).

注） 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

注） 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
2019/03/19	1	First Release
2022/03/31	2	Contents Revised
2022/06/01	3	Contents Revised

CMOS 32-Bit Microcontroller

TMPM3U0FSDMG

TMPM3U0FSDMG is a 32-bit RISC microprocessor series with an Arm® Cortex®-M3 microprocessor core. Features of the TMPM3U0FSDMG 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]
      - 32-bit multiplication (32 × 32 = 32bit) 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
   - On-chip RAM : 4Kbyte
   - On-chip FlashROM : 64Kbyte

3. 16-bit timer (TMRB) : 4 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - Input capture function
   - 16-bit PPG output
   - External trigger PPG output

4. Watchdog timer (WDT) : 1 channel

   Watchdog timer (WDT) generates a reset .

5. Power_On reset function (POR)

6. Voltage detect function (VLTD)

7. Oscillation frequency detect function (OFD)

8. Programmable motor driver (PMD) : 1channels
   - 3phase complementary PWM generator
   - Synchronous AD convert start trigger generator
   - Emergency protective function (EMG)

9. Encoder input circuit (ENC) : 1channels
   - Correspond to incremental encoder (AB / ABZ)
   - Rotation direction detection
   - Counter for absolute position detection
   - Comparator for position detection
   - Noise filter
   - 3 phase sensor input

10. General-purpose serial interface(SIO/UART) : 2channels
    - Either UART mode or synchronous mode can be selected (4byte FIFO equipped)(1channel)
    - UART mode (1channel)

11. Serial bus interface (I2C/SIO) : 1 channel

    Either I2C bus mode or synchronous mode can be selected.

12. 12 bit AD converter (ADC) : 1units (Analog input : 4 channel)
    - Start by the internal trigger : TMRB interrupt / PMD trigger
    - Constant conversion mode
    - AD monitoring 2ch
    - Conversion speed 2 μsec (@ADC conversion clock = 40 MHz

13. Input/ output ports (PORT) : 21 pins

    I/O pin : 21pins

14. Interrupt source
    - Internal 29 factors : The order of precedence can be set over 7 levels. (except the watchdog timer interrupt)
    - External 3 factors : The order of precedence can be set over 7 levels.

15. Standby mode

    Standby modes : IDLE, STOP

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

17. Endian

    Little endian

18. Internal high-speed oscillation circuit : 10 MHz

19. Maximum operating frequency : 40 MHz

20. Operating voltage range
    - DVDD5B = 4.5V to 5.5V(fsys=40MHz)

    All function operating

    - DVDD5B = 3.9V to 4.5V(fsys=40MHz)

    Without 12-bit ADC conversion accuracy, AC/DC Characteristics and Flash writing/erasing

21. Temperature range
    - -40 °C to 105 °C (except during Flash writing/erasing)
    - 0 °C to 70 °C (during Flash writing/erasing)

22. Package

    SSOP30 (5.6 mm x 9.7 mm , 0.65 mm pitch)

1.2 Block Diagram

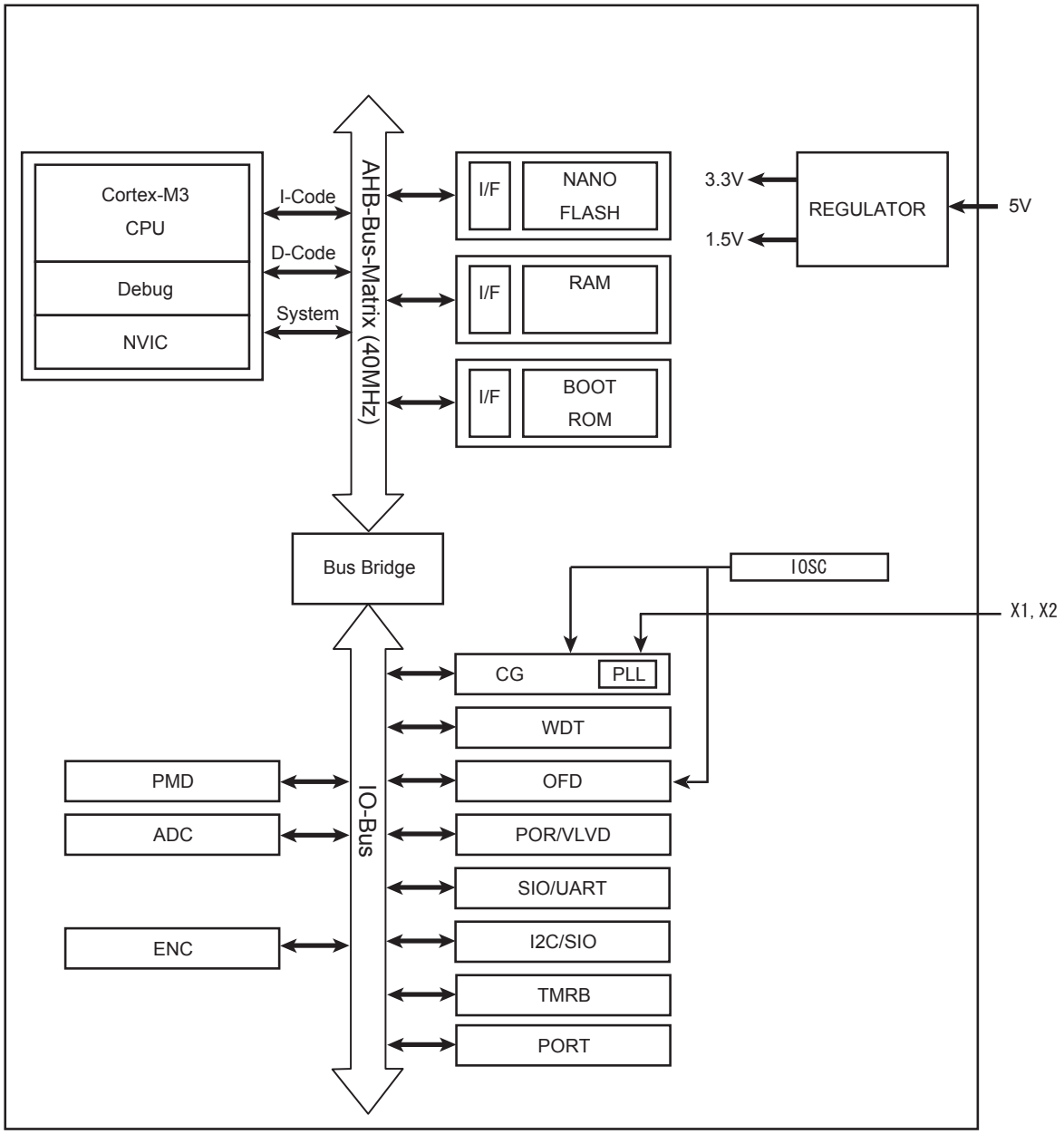


Figure 1-1 TMPM3U0FSDMG block diagram

1.3 Pin Layout (Top view)

The pin layout of TMPM3U0FSDMG is a figure below.

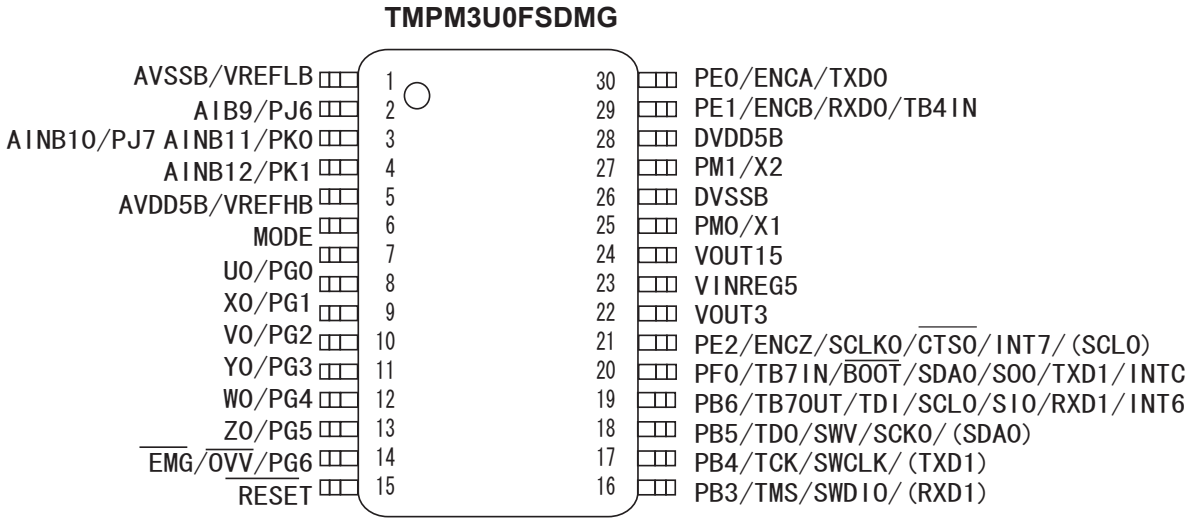


Figure 1-2 Pin Layout(SSOP30)

1.4 Pin names and Functions

Table 1-1 sorts the input and output pins of the TMPM3U0FSDMG by pin or port. Each table includes alternate pin names and functions for multi-function pins.

1.4.1 Sorted by Port

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

PORT	Type	Pin No.	Pin Name	Input / Out-put	Function
PORT B	Function/ Debug	16	PB3 TMS/SWDIO (RXD1)	I/O I/O I	I/O port Debug pin Receiving serial data
PORT B	Function/ Debug	17	PB4 TCK/SWCLK (TXD1)	I/O I/O O	I/O port Debug pin Sending serial data
PORT B	Function/ Debug	18	PB5 TDO/SWV SCK0 (SDA0)	I/O O I/O I/O	I/O port Debug pin Serial clock I2C data input/output
PORT B	Function/ Debug	19	PB6 TDI SI0/SCL0 TB7OUT INT6 RXD1	I/O I I/O O I I	I/O port Debug pin Serial data input / I2C clock input/output 16bit Timer output External Interrupt input pin Receiving serial data
PORT E	Function	30	PE0 TXD0 ENCA	I/O O I	I/O port Sending serial data A-phase input pin
PORT E	Function	29	PE1 RXD0 ENCB TB4IN	I/O I I I	I/O port Receiving serial data B-phase input pin Inputting the 16bit timer capture trigger
PORT E	Function	21	PE2 SCLK0 $\overline{CTS0}$ ENCZ INT7 (SCL0)	I/O I/O I I I/O	I/O port Serial clock input/ output Handshake input pin Z-phase input pin External Interrupt input pin I2C clock input/output
PORT F	Function/ Control	20	PF0 TB7IN SO0/SDA0 $\overline{BOOT}$ TXD1 INTC	I/O I I/O I O I	I/O port Inputting the 16bit timer capture trigger Serial data output / I2C data input/output BOOT mode pin. (note) This pin goes into single boot mode by sampling "Low" at the rise of a RESET signal. Sending serial data Interrupt input pin
PORT G	Function	8	PG0 U0	I/O O	I/O port U-phase output pin
PORT G	Function	9	PG1 X0	I/O O	I/O port X-phase output pin

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

PORT	Type	Pin No.	Pin Name	Input / Out-put	Function
PORT G	Function	10	PG2 V0	I/O O	I/O port V-phase output pin
PORT G	Function	11	PG3 Y0	I/O O	I/O port Y-phase output pin
PORT G	Function	12	PG4 W0	I/O O	I/O port W-phase output pin
PORT G	Function	13	PG5 Z0	I/O O	I/O port Z-phase output pin
PORT G	Function	14	PG6 $\overline{\text{EMG1}}$ $\overline{\text{OVV}}$	I/O I I	I/O port Emergency status detection input Overvoltage Detection input
PORT J	Function	2	PJ6 AINB9	I/O I	I/O port Analog input
PORT J	Function	3	PJ7 AINB10	I/O I	I/O port Analog input
PORT K	Function	4	PK0 AINB11	I/O I	I/O port Analog input
PORT K	Function	5	PK1 AINB12	I/O I	I/O port Analog input
PORTM	Function/ Clock	25	PM0 X1	I/O I	I/O port Connected to a high-speed oscillator
PORTM	Function/ Clock	27	PM1 X2	I/O O	I/O port Connected to a high-speed oscillator
-	Control	7	MODE	I	Mode pin (note) MODE pin must be connected to GND.
-	Function	15	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter (about 30ns (typical value))
-	PS	26	DVSSB	–	GND pin
-	PS	28	DVDD5B	–	Power supply pin
-	PS	23	VINREG5	–	Power supply pin
-	PS	24	VOUT15	–	Power supply pin
-	PS	22	VOUT3	–	Power supply pin
-	PS (Note 1)	1	AVSSB VREFLB	–	AD converter: GND pin Supplying the AD converter with a reference power supply.
-	PS (Note 2)	6	AVDD5B VREFHB	–	Supplying the AD converter with a power supply. Supplying the AD converter with a reference power supply.

Note 1: AVSS must be connected to GND even if the AD converter is not used.
Note 2: Must be connected to power supply even if AD converter is not used.

1.5 Pin Numbers and Power Supply Pins

Table 1-2 Pin Numbers and Power Supplies

Power supply	Voltage range	Pin No.	Pin name
DVDD5B	4.5 to 5.5V 3.9 to 4.5V(note)	28	PB,PE,PF,PG,PM $\overline{\text{RESET}}$,MODE
AVDD5B		6	PJ,PK
VINREG5		23	-

Note: Function operation has restriction. For detail , refer to the Electrical Characteristics.

Table 1-3 On-chip Regulator output pin

Pin name	Pin No.	remark
VOUT15	24	VOUT15 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit.
VOUT3	22	VOUT3 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit.

Note: VOUT15 and VOUT3 must be connected with the same value of capacitors. The IC outside can not have the power supply from VOUT15 and VOUT3.

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 TMPM3U0FSDMG.

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
TMPM3U0FSDMG	r2p1

2.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r2p1 are ETM™ and MPU. The following table shows the configurable options in the TMPM3U0FSDMG.

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

2.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

2.3.1 Number of Interrupt Inputs

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

TMPM3U0FSDMG has 32 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.

TMPM3U0FSDMG 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.

TMPM3U0FSDMG 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.

TMPM3U0FSDMG does not use this signal. To return from LOCKUP status, it is necessary to use non-mask-able interrupt (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, TMPM3U0FSDMG 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.

TMPM3U0FSDMG 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.

TMPM3U0FSDMG 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 TMPM3U0FSDMG does not use this function.

3. Memory Map

3.1 Memory Map

The memory maps for TMPM3U0FSDMG are based on the Arm Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM3U0FSDMG are mapped to the Code, SRAM and peripheral regions of the Cortex-M3 respectively. The special function register (SFR) means the control registers of all input/output ports and peripheral functions. The SRAM and SFR areas are all included in the bit-band region.

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

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

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

3.1.1 TMPM3U0FSDMG Memory Map

Figure 3-1 shows the memory map of the TMPM3U0FSDMG.

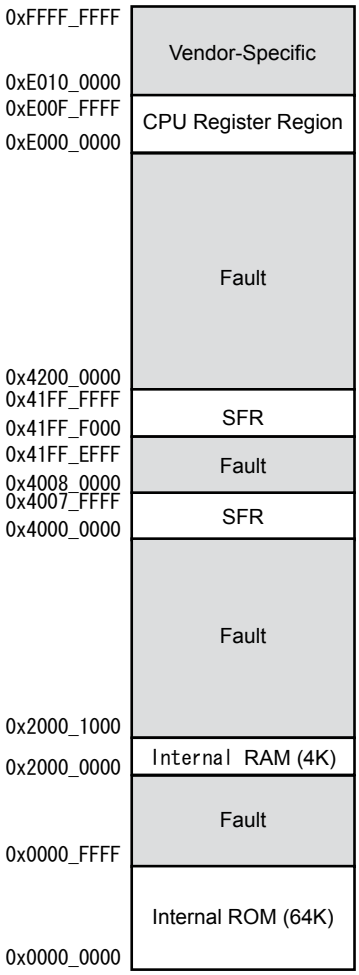


Figure 3-1 Memory Map

3.2 Details of SFR area

Table 3-1shows the details of the SFR area.

Do not access a reserved area in Table 3-1. See the chapter of each peripheral function for details.

Table 3-1 Details of SFR

Start Address	End Address	Peripheral
0x4000 _ 0000	0x4000 _033F	PORT
0x4000 _ 0340	0x4000_FFFF	Reserved
0x4001_0000	0x4001_01FF	TMRB
0x4001_0200	0x4001_03FF	Reserved
0x4001_0400	0x4001_043F	ENC
0x4001_0440	0x4001_FFFF	Reserved
0x4002_0000	0x4002_007F	I2C/SIO
0x4002_0080	0x4002_00FF	SIO/UART
0x4002_0100	0x4003_01FF	Reserved
0x4003_0200	0x4003_02FF	ADC
0x4003_0300	0x4003_0417	Reserved
0x4003_0418	0x4003_041F	Reserved
0x4003_0420	0x4003_FFFF	Reserved
0x4004_0000	0x4004_003F	WDT
0x4004_0040	0x4004_01FF	Reserved
0x4004_0200	0x4004_022F	CG
0x4004_0230	0x4004_02FF	Reserved
0x4004_0300	0x4004_030F	TRM
0x4004_0310	0x4004_07FF	Reserved
0x4004_0800	0x4004_083F	OFD
0x4004_0840	0x4004_08FF	Reserved
0x4004_0900	0x4004_093F	VLTD
0x4004_0940	0x4004_FFFF	Reserved
0x4005_0000	0x4005_01FF	Reserved
0x4005_0200	0x4005_047F	Reserved
0x4005_0480	0x4005_04FF	PMD
0x4005_0500	0x4005_FFFF	Reserved
0x4006_0000	0x4006_0007	DNF
0x4006_0008	0x4007_FFFF	Reserved
0x4008_0000	0x41FF_EFFF	Hard fault
0x41FF_F000	0x41FF_F03F	FLASH
0x41FF_F040	0x41FF_FFFF	Reserved

Table 3-2 Base Address List

Peripheral Name	Base Address	SFR
Prot B	0x4000 _0040	PORT
Port E	0x4000_0100	PORT
Port F	0x4000_0140	PORT
Port G	0x4000_0180	PORT

Table 3-2 Base Address List

Port J	0x4000_0240	PORT
Port K	0x4000_0280	PORT
Port M	0x4000_0300	PORT
TMRB 0	0x4001_0000	TMRB
TMRB 4	0x4001_0100	TMRB
TMRB 5	0x4001_0140	TMRB
TMRB 7	0x4001_01C0	TMRB
ENC 0	0x4001_0400	ENC
SBI 0	0x4002_0000	SBI
SIO UART 0	0x4002_0080	SIO / UART
SIO UART 1	0x4002_00C0	SIO / UART
ADC	0x4003_0200	ADC
WDT	0x4004_0000	WDT
CG	0x4004_0200	CG
OSCTRIM	0x4004_0300	TRM
OFD	0x4004_0800	OFD
LVD	0x4004_0900	LVD
PMD 1	0x4005_0480	PMD
DNF	0x4006_0000	DNF

4. Reset Operation

The following are sources of reset operation.

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

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

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

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

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

4.1 Cold Reset

When turning-on power, it is necessary to take a stable time of built-in regulator, built-in Flash memory and internal high-speed oscillator into consideration. TMPM3U0FSDMG has a function to insert a stable time automatical-ly.

4.1.1 Reset by power-on-reset circuit (not using $\overline{\text{RESET}}$ pin)

Once power voltage is beyond the release voltage of power-on-reset, power counter starts operation, and then after approximately 3.2ms internal reset signal is released.

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

Note: This device may start up in single boot mode, when the $\overline{\text{BOOT}}$ pin is "Low" level at power-on. Therefore when MCU starting in single mode, The $\overline{\text{BOOT}}$ pin must be at "High" level at power-on until reset release operation is completed.

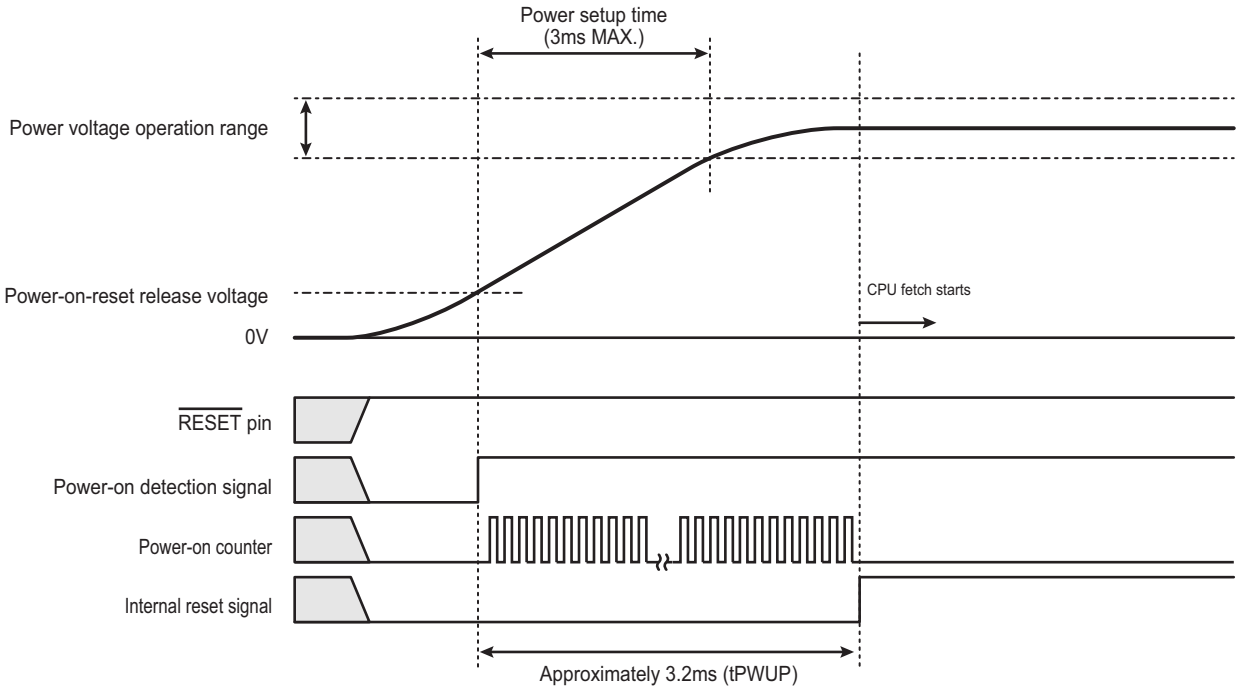


Figure 4-1 Reset Operation by Power-on Circuit

Note1: The above sequence is applied as well when restoring power.
Note2: The total rise time of the power supply from 0 V to the operating voltage range must be longer than 100 μs.

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

The reset using the $\overline{\text{RESET}}$ pin will be effective after the power-on counter finishes. And if $\overline{\text{RESET}}$ pin is set to "High" within 3.2ms after power-on reset signal becomes "High", the reset process will be the same as the power-on described in 4.1.1.

Note: This device may start up in single boot mode, when the $\overline{\text{BOOT}}$ pin is "Low" level at power-on. Therefore when MCU starting in single mode, The $\overline{\text{BOOT}}$ pin must be at "High" level at power-on until reset release operation is completed.

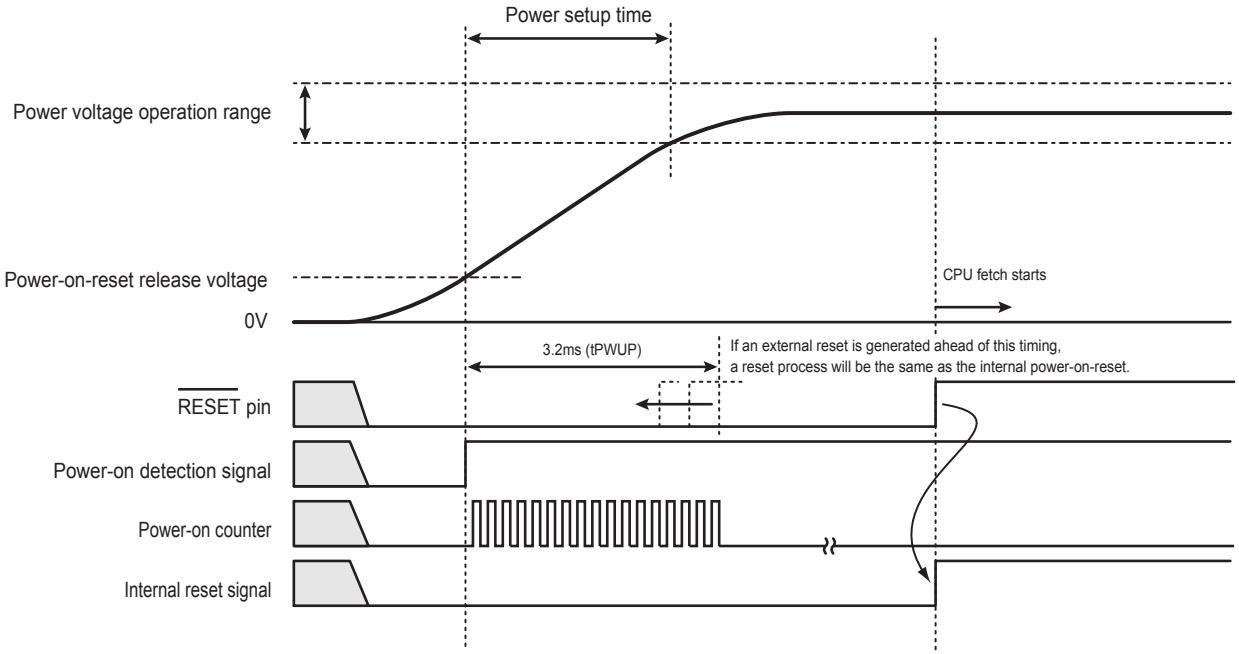


Figure 4-2 Reset Operation by $\overline{\text{RESET}}$ pin

Note1: The above sequence is applied as well when restoring power.
Note2: The total rise time of the power supply from 0 V to the operating voltage range must be longer than 100 µs.

4.2 Warm-up

4.2.1 Reset Duration

To do reset TMPM3U0FSDMG, the following condition is required; power supply voltage is in the operational range; RESET pin is kept "Low" at least for 12 system clocks by internal high frequency oscillator. After $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

4.3 After reset

After reset, the control register of Cortex-M3 and the peripheral function control register (SFR) are initialized. System debug component registers (FPB, DWT, and ITM) of the internal core, CGRSTFLG in the clock generator and FCSECBIT in the Flash related register are only initialized by cold reset.

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

5. Clock / Mode Control

5.1 Features

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

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

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

The clock/mode control block has the following functions:

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

In addition to NORMAL mode, the TMPM3U0FSDMG can operate in six types of low power mode to reduce power consumption according to its usage conditions.

5.2 Registers

5.2.1 Register List

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

Base Address = 0x4004_0200

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

5.2.2 CGSYSCR (System control register)

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

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

5.2.3 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				WUPSEL2	HOSCON	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	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-20	WUODR[11:0]	R/W	Specifies count time of the warm-up timer.
19	WUPSEL2	R/W	Clock source for Warm-up timer(WUP) 0: Internal (OSC2) 1: External (OSC1) Select source clock for warm-up timer between external oscillator (OSC1) and internal oscillator (OSC2).
18	HOSCON	R/W	Port M or external oscillator (X1/X2) (Note) 0: PORT M 1: External oscillator (X1/X2) Specifies Port M or X1/X2. When the external oscillator (OSC1) is used, Port M registers (PMCR/PMPUP/PMPDN/PMIE) should be disabled. After reset, the port M registers are disabled.
17	OSCSEL	R/W	Selection of high-speed oscillator 0: Internal (OSC2) 1: External (OSC1) Select high-speed oscillator between external oscillator (OSC1) and internal oscillator (OSC2). Confirm <OSCSEL> become "1" then halt the OSC2 immediately after switching over to OSC1. Do not select OSC2 again after switching to OSC1.
16	XEN2	R/W	High-speed oscillator2 (Internal) 0: Stop 1:Oscillation Specifies operation of the high-speed oscillator 2 (OSC2).
15-12	−	R/W	Write as "0".
11-10	−	R	Read as "0".
9	−	R/W	Write as "0".
8	XEN1	R/W	High-speed oscillator1 (External) 0: Stop 1:Oscillation Specifies operation of the high-speed oscillator 1 (OSC1).
7-4	−	R/W	Read as "0".
3	WUPSEL1	R/W	Clock source for Warm-up timer Write as "0".
2	PLLON	R/W	PLL operation 0: Stop 1: Oscillation Specifies operation of the PLL. It stops after reset.Setting the bit is required.

Bit	Bit Symbol	Type	Function
1	WUEF	R	Status of warm-up timer (WUP) 0: Warm-up completed. 1: Warm-up operation Enable to monitor the status of the warm-up timer.
0	WUEON	W	Operation of warm-up timer 0: don't care 1: Starting warm-up Enables to start the warm-up timer. (Note) Do not write "1" to <WUEON>, at the setting of returning from stop mode with automatic warming-up. When warming-up is started by software (<WUOEN> = "1"), please monitor <WUEF> and confirm warming-up is completed. After <WUEN> turn to "0" operation mode can be changed to stop mode.

Note: When the <HOSCON> is set to "1", the all registers for Port M can not be accessed and the read data from these registers are always "0". If one of the Port M registers except PMDATA and PMOD is not equal to the initial value, the <HOSCON> can not be set to "1".

5.2.4 CGSTBYCR (Standby control register)

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

Bit	Bit Symbol	Type	Function
31-18	−	R	Read as "0".
17	−	R/W	Write as "0".
16	DRVE	R/W	Pin status in STOP mode 0: Inactive 1: Active
15-10	−	R	Read as "0".
9	−	R/W	Write as "0".
8	RXEN	R/W	High-speed oscillator operation after releasing the STOP mode. Write as "1".
7-3	−	R	Read as "0".
2-0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP 010: Reserved 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved To enter the STOP mode, disable the oscillation (OSC1 or OSC2) which is unused as system clock.

5.2.5 CGPLLSEL (PLL Selection Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PLLSET1				-	PLLSET0		
After reset	1	1	0	1	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PLLSET0							PLLSEL
After reset	0	0	0	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-12	PLLSET1[3:0]	R/W	Write as "1101".
11	−	R	Read as "0".
10-1	PLLSET0[9:0]	R/W	PLL multiple setting Write as a follow setting when it is used 8MHz external clock. 5 PLL: "01_0001_0011". Write as a follow setting when it is used 10MHz external clock. 4 PLL: "01_0000_1111". Note: Do not set any values other than those shown above setting.
0	PLLSEL	R/W	Use of PLL 0: fosc use 1: PLL Use Specifies use or disuse of the clock multiplied by the PLL. "fosc" is automatically set after reset. Resetting is required when using the PLL.

5.3 Clock control

5.3.1 Clock Type

Each clock is defined as follows :

fosc1	: Clock input from external high-speed oscillator (X1 and X2)
fosc2	: Clock input from internal high-speed oscillator
fosc	: High-speed clock specified by CGOSCCR<OSCSEL>
f_{PLL}	: Clock octupled by PLL (x 5 or 4)
fc	: Clock specified by CGPLLSEL<PLLSEL> (high-speed clock)
fgear	: Clock specified by CGSYSCR<GEAR[2:0]>
fsys	: The same clock as fgear (system clock)
fperiph	: Clock specified by CGSYSCR<FPSEL>
φT0	: Clock specified by CGSYSCR<PRCK[2:0]> (Prescaler clock)

The high-speed clock fc and the prescaler clock φT0 are dividable as follows.

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

5.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

High-speed oscillator 1 (External)	: Stop
High-speed oscillator 2 (Internal)	: Oscillating
PLL (Phase locked loop circuit)	: Stop
High-speed clock gear	: fc (no frequency dividing)

Reset operation causes all the clock configurations to be the same as f_{OSC2}.

$f_C = f_{OSC2}$

$f_{SYS} = f_C$ (= f_{OSC2})

$f_{periph} = f_C$ (= f_{OSC2})

$\varphi T0 = f_{periph}$ (= f_{OSC2})

5.3.3 Clock system Diagram

Figure 5-1 shows the clock system diagram.

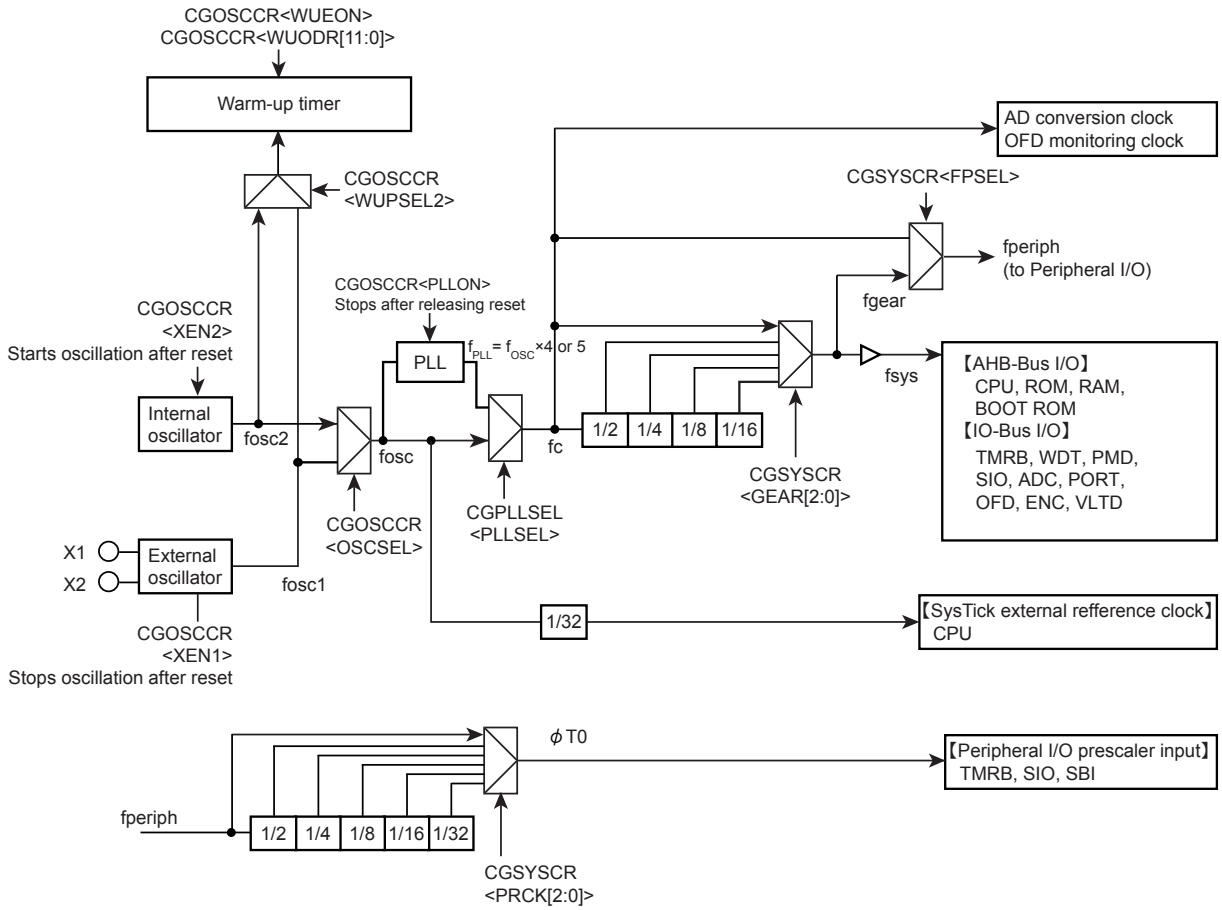


Figure 5-1 Clock Block Diagram

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

5.3.4 Clock Multiplication Circuit (PLL)

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

The PLL is disabled after reset. To enable the PLL, set "1" to the CGOSCCR<PLLON> bit and set "1" to the CGPLLSEL<PLLSEL>. Then f_{PLL} clock output is quadruple or quintuple of the high-speed oscillator (fosc).

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

5.3.4.1 Stability time

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

When the <PLLON> is set to "1" and operation starts, it is necessary to take approximately 200μs as the Lock-up time.

The <PLLON> is first made "0" when the multiplying value is changed and PLL is stopped. When the multipling <PLLSEL> value is changed, the <PLLON> is set to "1" after approximately 100μs elapses as initialization time of PLL, and the state of PLL starts. Afterwards, please secure the Lock-up time as PLL stability time.

5.3.4.2 The sequence of PLL setting

The following shows PLL setting sequence after reset.

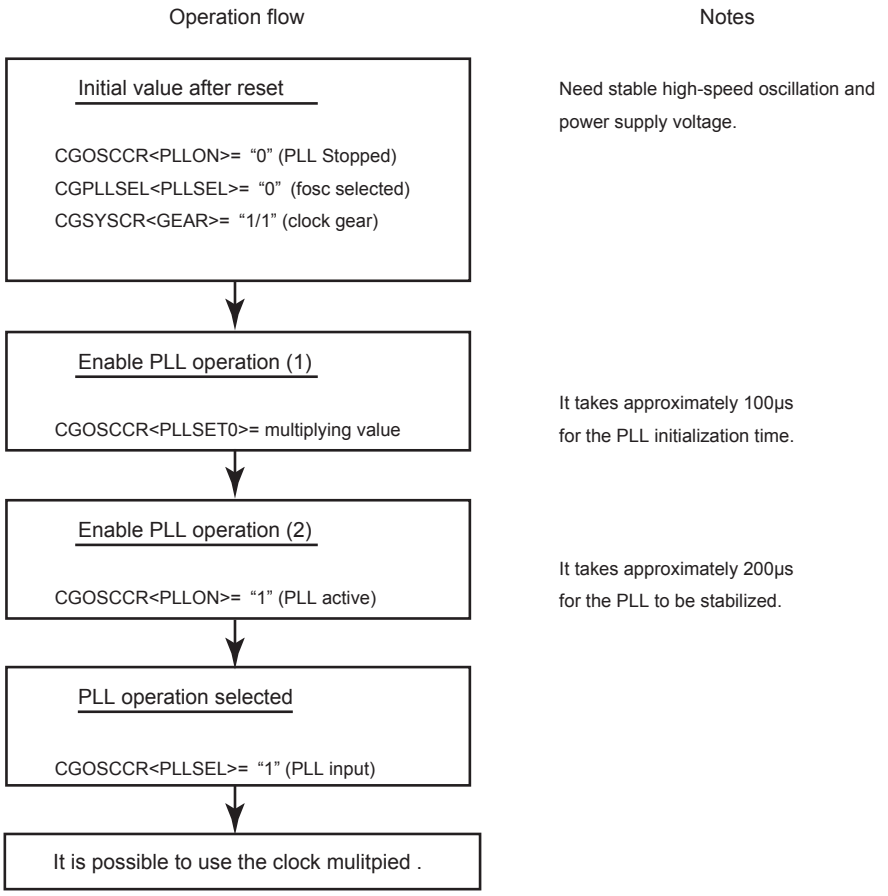


Figure 5-2 PLL setting sequence after reset

Note: When you stop PLL, please check that it is the register CGPLLSEL<PLLSEL> = "0" after setting up the CGPLLSEL<PLLSEL> = "0". Then, please set up CGOSCCR<PLLON> = "0" (PLL stoped).

Note: After setting PLL multiplying value, to keep CGOSCCR<PLLON>="0"(PLL stop) over 100µs is needed as the PLL initializing stable time.

5.3.5 Warm-up function

The warm-up function secures the stability time for the oscillator and the PLL with the warm-up timer. The warm-up function is used when returning from STOP mode. For detail function, describes in "5.6.6 Warm-up".

Note: Do not shift to STOP mode, during operating warm-up timer.

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

In STOP modes, the PLL is disabled. When returning from these modes, configure the warm-up time in consideration of the stability time of the PLL and the internal oscillator.

How to configure the warm-up function.

1. Specify the count up clock

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

   Specify the count up clock for the warm-up counter in the CGOSCCR<WUPSEL1> and <WUPSEL2> bit. Write "0" to <WUPSEL1> and write "0" or "1" to <WUPSEL2>. "0" specifies internal oscillator and "1" specifies external oscillator.

2. Specify the warm-up counter value

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

   The following shows the warm-up setting and example.

$$\text{Warm-up cycles} = \frac{\text{Setting value of warm-up time}}{\text{Input cycle by frequency(s)}}$$

<example 1>Setting 5 ms of warm-up time with 8MHz oscillator

$$\frac{\text{Setting value of warm-up time}}{\text{Input cycle by frequency(s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40{,}000\text{cycles} = 0\text{x}9\text{C}40$$

Drop the last 4 bits, set 0x9C4 into the CGOSCCR<WUPT[11:0]>.

3. Confirm the start and completion of warm-up

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

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

The following shows the warm-up setting.

<example> Securing the stability time for the PLL (fc = fosc1)

CGOSCCR<WUPSEL1> = "0"	: Write "0" to CGOSCCR<WUPSEL1>
CGOSCCR<WUPSEL2> = "1"	: Specify the clock source for warm-up timer
CGOSCCR<WUODR[11:0]> = "0x9C4"	: Warm-up time setting
Refer to 5.3.6 for the procedure of switching over from the internal oscillator to the external oscillator.	
CGOSCCR<WUEON>="1"	: Enable warm-up counting (WUP)
Read CGOSCCR<WUEF>	: Wait until the state becomes "0" (warm-up is finished)

5.3.6 System Clock

The TMPM3U0FSDMG offers high-speed clock as system clock. System clock is selectable from internal oscillator or external oscillator. After reset, internal oscillator is enabled and external oscillator is disabled. The high-speed clock is dividable.

- Input frequency from X1 and X2 : 8 MHz to 10MHz
- Internal oscillator frequency : 10MHz
- Clock gear : 1/1, 1/2, 1/4, 1/8, 1/16 (after reset : 1/1)

Table 5-1 Range of high-speed frequency (unit : MHz)

Input freq.		Min. operating freq.	Max. operating freq.	After reset (PLL = OFF, CG = 1/1)	Clock gear (CG) : PLL = ON					Clock gear (CG) : PLL = OFF				
					1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
OSC1	8	1	40	8	40	20	10	5	2.5	8	4	2	1	-
	10			10	40	20	10	5	2.5	10	5	2.5	1.25	-
OSC2	10			10	40	20	10	5	2.5	10	5	2.5	1.25	-

Note 1: PLL=ON/OFF setting: available in CGOSCCR<PLLON>.

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

Note 3: ."-" : Reserved

Note 4: Do not use 1/16 when SysTick is used.

The following are the procedure of switching over from the internal oscillator to the external oscillator.

1. Disables port M registers (PMCR/PMPUP/PMPDN/PMIE). After reset, these registers are disabled.
2. CGOSCCR<WUODR[11:0]> = "Warm-up time" : Set Warm-up time.
3. CGOSCCR<HOSCON> = "1" : Switch over from the port M to oscillator connection pins..
4. CGOSCCR<XEN1> = "1" : Enable the external oscillator.
5. CGOSCCR<WUPSEL2> = "1" : Specify the external oscillator clock as source clock for warm-up counter.
6. CGOSCCR<WUEON>="1" : Enable warm-up counting (WUP)
   Read CGOSCCR<WUEF> : Wait until the state becomes "0" (warm-up is finished)
7. CGOSCCR<OSCSEL> = "1" : Switch the system clock to the external oscillator.
8. Read CGOSCCR<OSCSEL> : Confirm CGOSCCR[17]<OSCSEL> become "1". (External oscillator is selected.)
9. CGOSCCR<XEN2> = "0" : Internal oscillator is disabled.

With setting CGOSCCR<HOSCON> to "1", rewriting the portM registers (PMDATA/PMCR/PMOD/PMPUP/PMPDN/PMIE) are prohibited.

5.3.7 Prescaler Clock Control

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

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

5.4 Modes and Mode Transitions

5.4.1 Mode Transitions

The NORMAL mode use the high-speed clock for the system clock .

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

Figure 5-3 shows mode transition diagram.

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

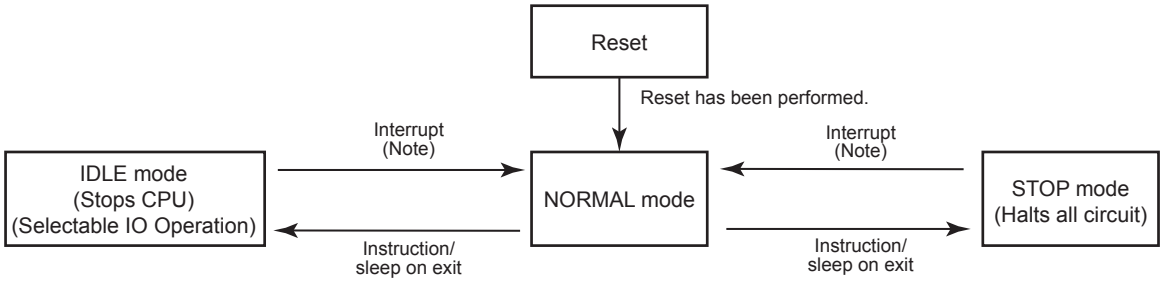


Figure 5-3 Mode Transition Diagram

Note: The warm-up is needed. The warm-up time must be set in NORMAL mode before changing to STOP mode. Regarding warm-up time, refer to "5.6.6 Warm-up".

5.5 Operation Mode

As an operation mode, NORMAL is available. The features of NORMAL mode are described in the following section.

5.5.1 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.

5.6 Low Power Consumption Modes

The TMPM3U0FSDMG has two low power consumption modes: IDLE and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction.In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

Note 1: The TMPM3U0FSDMG 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 TMPM3U0FSDMG does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

The features of each mode are described as follows.

5.6.1 IDLE Mode

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

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

- 16-bit timer / event counter (TMRB)
- Serial channel (SIO/UART)
- Serial bus interface (I2C/SIO)
- Watchdog timer (WDT)

Note: WDT should be stopped before entering IDLE mode.

5.6.2 STOP mode

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

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

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 5-2 shows the pin status in the STOP mode.

Table 5-2 Pin States in the STOP mode

	Pin name	I/O	<DRVE> = 0	<DRVE> = 1
Not port	$\overline{\text{RESET}}$, MODE	Input only	o	
	VOUT15, VOUT3	Output only	o	
Port	X1	Input only	×	
	X2	Output only	"High" level output	
	TMS TCK TDI	Input	o	
	TDO/SWV	Output	Enabled when data is valid. Disabled when data is invalid.	
	SWCLK	Input	o	
	SWDIO	Input	o	
		Output	Enabled when data is valid. Disabled when data is invalid.	
	U0, V0, W0, X0, Y0,Z0	Output	Enabled when data is valid. Disabled when data is invalid.	
	INT6, INT7, INTC	Input	o	
	Other function pins other than the above or the ports that are used as general purpose ports.	Input	×	o
		Output	×	o

o : Input or output enabled.

× : Input or output disabled.

5.6.3 Low power Consumption Mode Setting

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

Table 5-3 shows the mode setting in the <STBY[2:0]>.

Table 5-3 Low power consumption mode setting

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

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

5.6.4 Operational Status in Each Mode

Table 5-4 shows the operational status in each mode.

Table 5-4 Operational Status in Each Mode

Block	NORMAL	IDLE	STOP
Processor core	o	×	×
I/O port	o	o	* (Note1)
PMD	o	o	×
ENC	o	o	×
OFD	o	o	×
ADC	o	o	×
SIO	o	ON/OFF selectable for each module	×
SBI	o		×
TMRB	o		×
WDT	o		×
VLTD	o	o	o (Note2)
POR	o	o	o (Note2)
DNF	o	o	×
CG	o	o	×
PLL	o	o	×
High-speed oscillator (fc)	o	o	×

o : Operating

× : Stopped

Note 1: It depends on CGSTBYCR<DRVE>.

Note 2: The blocks are not stopped even though the clock is halted.

5.6.5 Releasing the Low Power Consumption Mode

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

Details are shown in Table 5-5.

Table 5-5 Release Source in Each Mode

Low power consumption mode			IDLE (programable)	STOP
Release source	Interrupt	INT6, 7, C (Note1)	o	o
		INTRX0, 1, INTTX0, 1	o	×
		INTVCNB	o	×
		INTEMG1	o	×
		INTOVV1	o	×
		INTADBPDB	o	×
		INTTB00, 40, 50, 70 INTTB01, 41, 51, 71	o	×
		INTPMD1	o	×
		INTCAP00, 50, 70 INTCAP01, 51, 71	o	×
		INTADBCPA, INTADBCPB	o	×
		INTADBSFT	o	×
		INTADBTMR	o	×
		INTENC0	o	×
		INTSBI0	o	×
	NMI (INTWDT)		o	×
	RESET ($\overline{\text{RESET}}$ pin)		o	o

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

Note 1: To release the low power consumption mode by using the level mode interrupt, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt handling from starting properly.
Note 2: For shifting to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified for wake up.
Note 3: Refer to "5.6.6 Warm-up" about warm-up time.

- 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 STOP modes. And the digital noize filter circuit should be set to disable as well.

- Release by Non-Maskable Interrupt (NMI)

  There is a watchdog timer interrupt (INTWDT) as a non-maskable interrupt source. INTWDT can only be used in the IDLE mode.

Note: Notice that the WDT can not be cleared by CPU operation in IDLE mode.

- Release by reset

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

- Release by SysTick interrupt

  SysTick interrupt can only be used in IDLE mode.

Refer to "Interrupts" for detail.

5.6.6 Warm-up

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

In the mode transition from STOP to the NORMAL, the warm-up counter is activated automatically. And then the system clock output is started after the elapse of configured warm-up time. It is necessary to set a oscillator to be used for warm-up in the CGOSCCR<WUPSEL1><WUPSEL2> (Note1) and to set a warm-up time in the CGOSCCR<WUODR> before executing the instruction to enter the STOP mode.

Note 1: Always set CGOSCCR<WUPSEL1> to "0".

Note 2: In STOP modes, the PLL is disabled. When returning from these mode, configure the warm-up time in consideration of the stability time of the PLL and the internal oscillator. It takes approximately 200μs for the PLL to be stabilized.

Note 3: Do not write "1" to CGOSCCR<WUEON> bit, at the setting of returning from low consumption mode with automatic warming-up.

Table 5-6 shows whether the warm-up setting of each mode transition is required or not.

Table 5-6 Warm-up setting in mode transition

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

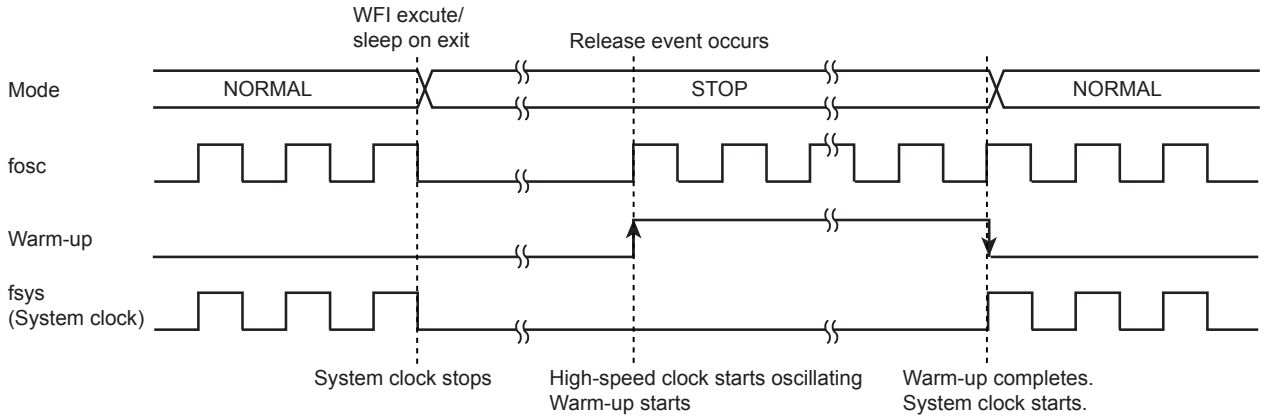
5.6.7 Clock Operation in Mode Transition

The clock operation in mode transition are described Chapter 5.6.7.1.

5.6.7.1 Transition of operation modes : NORMAL → STOP → NORMAL

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

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



6. Internal High-speed Oscillation Adjustment Function

TMPM3U0FSDMG has the internal high-speed oscillation adjustment function.

6.1 Structure

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

Figure 6-1 shows the function configuration.

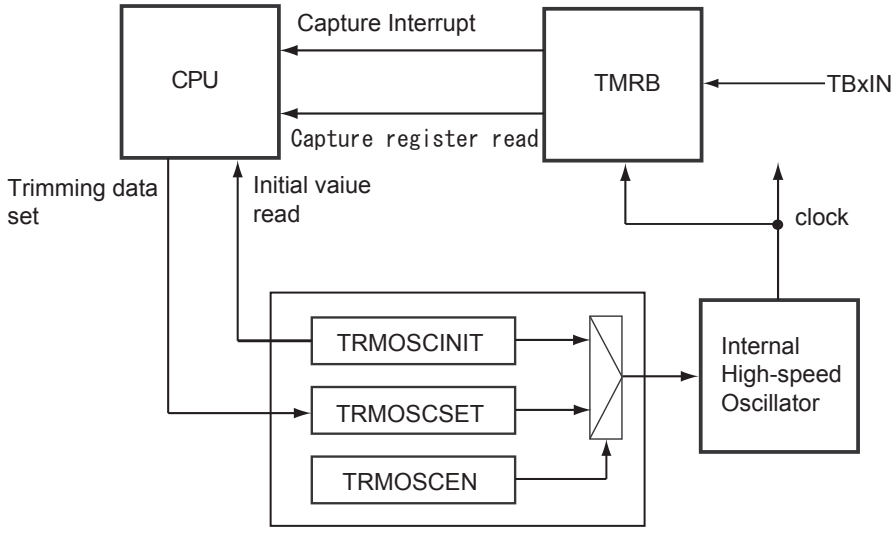


Figure 6-1 Function block diagram

6.2 Registers

6.2.1 Register list

The control registers and its addresses are as follows.

Base Address = 0x4004 _ 0300

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

6.2.2 TRMOSCPRO (Protect register)

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

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

6.2.3 TRMOSCEN (Enable register)

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

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

6.2.4 TRMOSCINIT (Initial trimming value monitor register)

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

Bit	Bit Symbol	Type	Function
31-14	−	R	Read as "0".
13-8	TRIMINITC [5:0]	R/W	Initial coarse trimming value Enables to monitor initial coarse trimming value.
7-4	−	R	Read as "0".
3-0	TRIMINITF[3:0]	R/W	Initial fine trimming value Enables to monitor initial fine trimming value.

For details about the specific setting and adjustment value of coarse trimming and fine trimming, refer to "Table 6-1 Adjustment range".

6.2.5 TRMOSCSET (Trimming value setting register)

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

Bit	Bit Symbol	Type	Function
31-14	−	R	Read as "0".
13-8	TRIMSETC [5:0]	R/W	Coarse trimming value setting Sets the coarse trimming value.
7-4	−	R	Read as "0".
3-0	TRIMSETF[4:0]	R/W	Fine trimming value setting Sets the fine trimming value.

For details about the specific setting and adjustment value of coarse trimming and fine trimming, refer to "Table 6-1 Adjustment range".

6.3 Operational Description

6.3.1 Outline

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

The value setting before shipping can be checked with TRMOSCINIT<TRIMINITC> and <TRIMINITF>. When the value changing, set a new value to TRMOSCSET<TRIMSETC> and <TRIMSETF>. By setting "1" to TRMOSCEN<TRIMEN>, a setting value of the internal oscillator will be changed.

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

6.3.2 Adjustment range

In the coarse trimming, −57.6% to +55.8% adjustment by 1.8%-step is feasible. In the fine trimming, −2.4% to +2.1% adjustment by 0.3%-step is feasible. Table 6-1 shows a adjustment range.

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

Table 6-1 Adjustment range

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

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

6.3.3 Internal Oscillation Frequency Measurement using TMRB

To measure a frequency of high-speed oscillator, the pulse width measurement function of TMRB can be used. First, choose an internal oscillator as a prescaler clock ΦT0 of TMRB. Second, input a pulse from TBxIN. Third, capture an up-counter value at the rising edge of the pulse using the capture function. Finally, determine the adjustment value using a difference between a frequency of TBxIN calculated with capture value and the actual frequency.

7. Exceptions

This chapter describes features, types and handling of exceptions.

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

7.1 Overview

There are two types of exceptions: those that are generated when some error condition occurs or when an instruction to generate an exception is executed; and those that are generated by hardware, such as an interrupt request signal from an external pin or peripheral function.

All exceptions are handled by the Nested Vectored Interrupt Controller (NVIC) in the CPU according to the respective priority levels. When an exception occurs, the CPU stores the current state to the stack and branches to the corresponding interrupt service routine (ISR). Upon completion of the ISR, the information stored to the stack is automatically restored.

7.1.1 Exception types

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

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

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

7.1.2 Handling Flowchart

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

Each step is described later in this chapter.

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

7.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

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

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

An interrupt request is generated from an external interrupt pin or peripheral function.For interrupts that are used for releasing a standby mode, relevant settings must be made in the clock generator. For details, refer to "7.5 Interrupts".

(2) Exception detection

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

Table 7-1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 7-1 Exception Types and Priority

No.	Exception type	Priority	Description
1	Reset	−3 (highest)	Reset pin, WDT, POR, VLTD, OFD or SYSRETREQ
2	Non-Maskable Interrupt	−2	WDT
3	Hard Fault	−1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4	Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) (Note 1) Instruction fetch from the Execute Never (XN) region
5	Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map
6	Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution
7~10	Reserved	−	
11	SVCall	Configurable	System service call with SVC instruction
12	Debug Monitor	Configurable	Debug monitor when the CPU is not faulting
13	Reserved	−	
14	PendSV	Configurable	Pendable system service request
15	SysTick	Configurable	Notification from system timer
16~	External interrupt	Configurable	External interrupt pin or peripheral function (Note2)

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

Note 2: **External interrupts have different sources and numbers in each product. For details, see"7.5.1.5 List of Interrupt Sources".**

(3) Priority setting

- Priority level

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

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

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

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

- Priority grouping

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

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

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

Table 7-2 Priority grouping setting

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

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

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

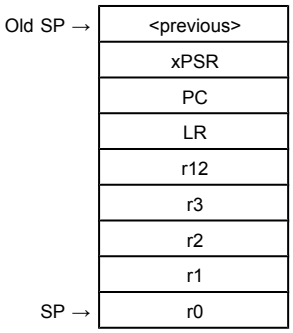
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

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

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

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



(2) fetching an ISR

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

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

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

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

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

7.1.2.3 Executing an ISR

An ISR performs necessary processing for the corresponding exception. ISRs must be prepared by the user.

An ISR may need to include code for clearing the interrupt request so that the same interrupt will not occur again upon return to normal program execution.

For details about interrupt handling, see "7.5 Interrupts".

If a higher priority exception occurs during ISR execution for the current exception, the CPU abandons the currently executing ISR and services the newly detected exception.

7.1.2.4 Exception exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

(2) Exception exit sequence

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

- Pop eight registers

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

- Load current active interrupt number

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

- Select SP

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

7.2 Reset Exceptions

Reset exceptions are generated from the following six sources.

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

- External reset pin

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

- Reset exception by POR

  Please refer the chapter "POR Power on Reset circuit" for detail.

- Reset exception by VLTD

  Please refer the chapter "VLTD Voltage Detection Circuit" for detail.

- Reset exception by OFD

  Please refer the chapter "OFD Oscillation Frequency Detector" for detail.

- Reset exception by WDT

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

- Reset exception by SYSRESETREQ

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

7.3 Non-Maskable Interrupts (NMI)

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

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

7.4 SysTick

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

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

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

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

7.5 Interrupts

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

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

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

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

7.5.1 Interrupt Sources

7.5.1.1 Interrupt route

Figure 7-1 shows an interrupt request route.

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

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

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

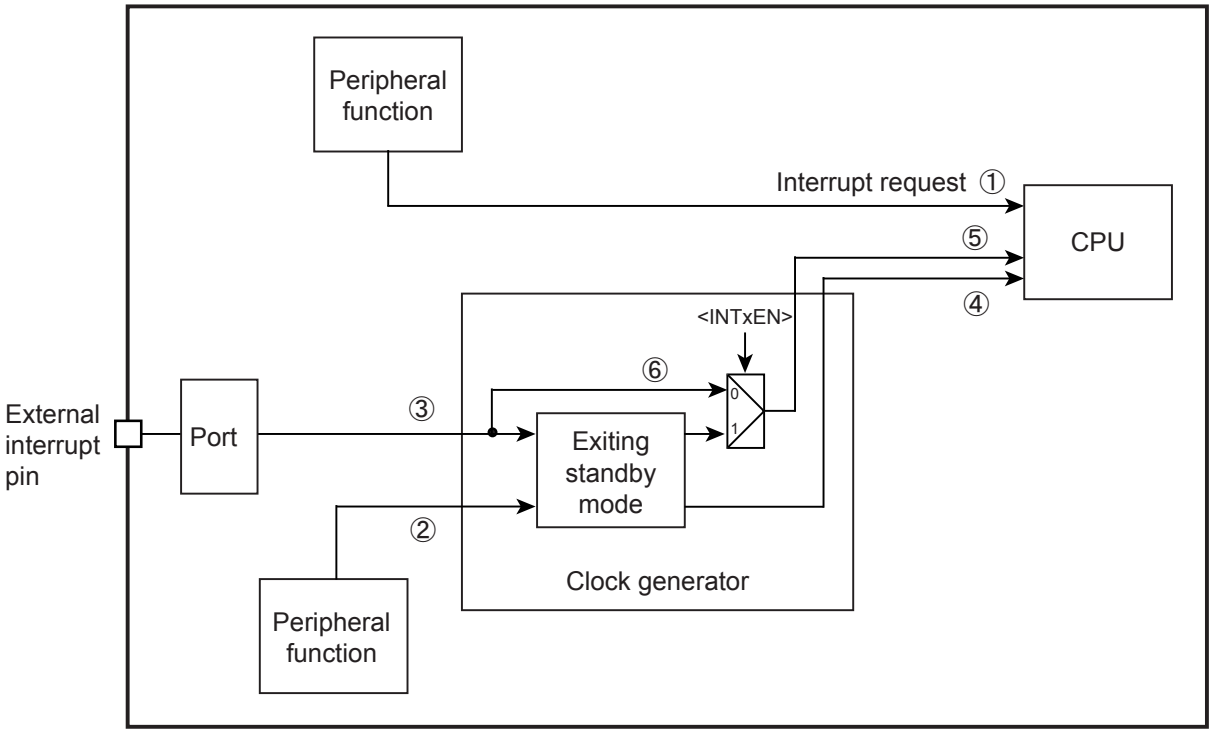


Figure 7-1 Interrupt Route

7.5.1.2 Generation

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

- From external pin

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

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

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

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

7.5.1.3 Transmission

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

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

7.5.1.4 Precautions when using external interrupt pins

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

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

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

7.5.1.5 List of Interrupt Sources

Table 7-3 shows the list of interrupt sources.

Table 7-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INTRX0	Serial reception (channel0)		
1	INTTX0	Serial transmit (channel0)		
2	INTRX1	Serial reception (channel1)		
3	INTTX1	Serial transmit (channel1)		
4	Reserved	Reserved		
5	INTEMG1	PMD1 EMG interrupt		
6	INTOVV1	PMD1 OVV interrupt		
7	INTADBPDB	ADCB conversion triggered by PMD1 is finished		
8	INTTB00	16bit TMRB0 compare match detection 0/ Over flow		
9	INTTB01	16bit TMRB0 compare match detection 1		
10	INTTB40	16bit TMRB4 compare match detection 0/ Over flow		
11	INTTB41	16bit TMRB4 compare match detection 1		
12	INTTB50	16bit TMRB5 compare match detection 0/ Over flow		
13	INTTB51	16bit TMRB5 compare match detection 1		
14	INTPMD1	PMD1 PWM interrupt		
15	INTCAP00	16bit TMRB0 input capture 0		
16	INTCAP01	16bit TMRB0 input capture 1		
17	INTCAP50	16bit TMRB5 input capture 0		
18	INTCAP51	16bit TMRB5 input capture 1		
19	INT6	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGA
20	INT7	Interrupt Pin		
21	INTADBCPA	ADCB conversion monitoring function interrupt A		
22	INTADBCPB	ADCB conversion monitoring function interrupt B		
23	INTADBSFT	ADC unit B conversion started by software is finished		
24	INTADBTMR	ADC unit B conversion triggered by timer is finished		
25	INTENC0	Encoder input0 interrupt		
26	INTTB70	16bit TMRB7 compare match detection 0/ Over flow		
27	INTTB71	16bit TMRB7 compare match detection 1		
28	INTCAP70	16bit TMRB7 input capture 0		
29	INTCAP71	16bit TMRB7 input capture 1		
30	INTC	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGA
31	INTSBI0	Serial bus interface		

7.5.1.6 Active level

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

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

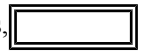
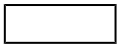
If an interrupt source is used for clearing a standby mode, setting the relevant clock generator register is also required. Enable the CGIMCGn<INTmEN> bit and specify the active level in the CGIMCGn<EMCGm> bits. You must set the active level for interrupt requests from each peripheral func-tion as shown in Table 7-3

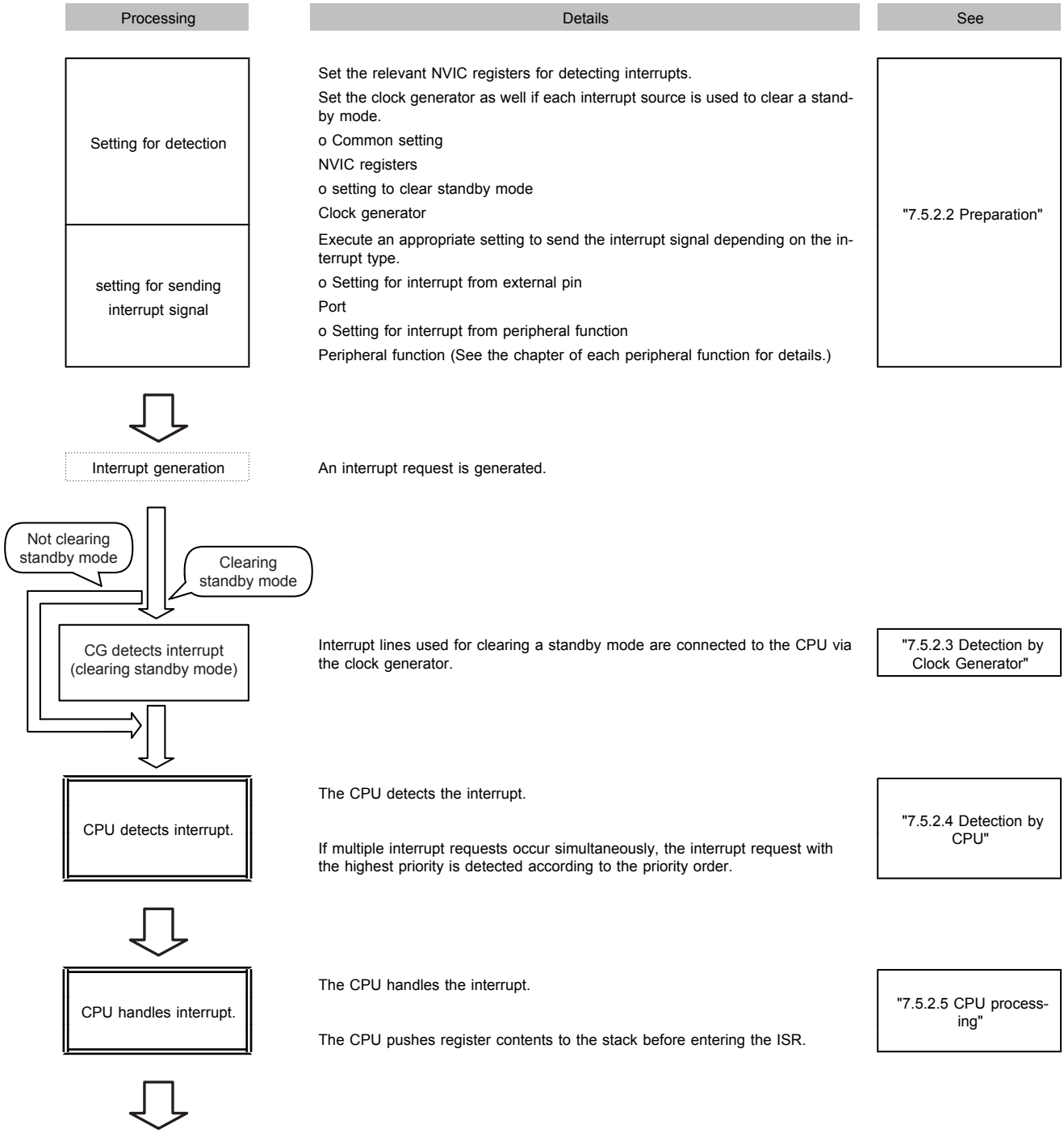
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

7.5.2 Interrupt Handling

7.5.2.1 Flowchart

The following shows how an interrupt is handled.

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



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

7.5.2.2 Preparation

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

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

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

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

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

(1) Disabling interrupt by CPU

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

Use "MSR" instruction to set this register.

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

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

(2) CPU registers setting

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note: m: corresponding bit

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "7.6.3.2 CGICRCG (CG Interrupt Request Clear Register)" for each value.

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

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

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

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

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

7.5.2.3 Detection by Clock Generator

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

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

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

7.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

7.5.2.5 CPU processing

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

7.5.2.6 Interrupt Service Routine (ISR)

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

(1) Pushing during ISR

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

Push the contents of other registers if needed.

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

(2) Clearing an interrupt source

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

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

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

7.6 Exception / Interrupt-Related Registers

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

7.6.1 Register List

NVIC registers — Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Reserved	0x0104
Reserved	0x0108
Interrupt Clear-Enable Register 1	0x0180
Reserved	0x0184
Reserved	0x0188
Interrupt Set-Pending Register 1	0x0200
Reserved	0x0204
Reserved	0x0208
Interrupt Clear-Pending Register 1	0x0280
Reserved	0x0284
Reserved	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 register — Base Address = 0x4004_0200

Register name		Address
CG Interrupt Request Clear Register	CGICRCG	0x0014
NMI Flag Register	CGNMIFLG	0x0018
Reset Flag Register	CGRSTFLG	0x001C
CG Interrupt Mode Control Register A	CGIMCGA	0x0020

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

7.6.2 NVIC Registers

7.6.2.1 SysTick Control and Status Register

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

Bit	Bit Symbol	Type	Function
31-17	−	R	Read as 0.
16	COUNTFLAG	R/W	0: Timer not counted to 0 1: Timer counted to 0 Returns "1" if timer counted to "0" since last time this was read. Clears on read of any part of the SysTick Control and Status Register.
15-3	−	R	Read as 0.
2	CLKSOURCE	R/W	0: External reference clock (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> by 32 is used as external reference clock.

7.6.2.2 SysTick Reload Value Register

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

Bit	Bit Symbol	Type	Function
31-24	−	R	Read as 0,
23-0	RELOAD	R/W	Reload value Set the value to load into the SysTick Current Value Register when the timer reaches "0".

7.6.2.3 SysTick Correct Value Register

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

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

7.6.2.4 SysTick Calibration Value Register

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

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

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

7.6.2.5 Interrupt Set-Enable Register 1

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

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

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

7.6.2.6 Interrupt Clear-Enable Register 1

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

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

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

7.6.2.7 Interrupt Set-Pending Register 1

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

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

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

7.6.2.8 Interrupt Clear-Pending Register 1

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

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

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

7.6.2.9 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

The number of bits to be used for assigning a priority varies with each product. This product uses three bits for assigning a priority.

The following shows the fields of the Interrupt Priority Registers for interrupt numbers 0 to 3. The Interrupt Priority Registers for all other interrupt numbers have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.

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

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

7.6.2.10 Vector Table Offset Register

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

Bit	Bit Symbol	Type	Function
31-7	TBLOFF	R/W	Offset value Set the offset value from the top of the space specified in TBLOFF. The offset must be aligned based on the number of exceptions in the table. This means that the minimum alignment is 32 words that you can use for up to 16 interrupts. For more interrupts, you must adjust the alignment by rounding up to the next power of two.
6-0	−	R	Read as 0,

7.6.2.11 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 (Written) / VECTKEYSTAT (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 endianl
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 it the responsibility of the application to reinitialize the stack.
0	VECTRESET	R/W	System Reset bit 1: reset system. 0: do not reset system. Resets the system, with the exception of debug components (FPB, DWT and ITM) by setting "1" and this bit is also zero cleared.

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

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

7.6.2.12 System Handler Priority Register

Each exception is provided with eight bits of a System Handler Priority Register.

The following shows the addresses of the System Handler Priority Registers corresponding to each exception.

	31 24	23 16	15 8	7 0
0xE000_ED18	PRI_7	PRI_6 (Usage Fault)	PRI_5 (Bus Fault)	PRI_4 (Memory Management)
0xE000_ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12 (Debug Monitor)

The number of bits to be used for assigning a priority varies with each product. This product uses three bits for assigning a priority.

The following shows the fields of the System Handler Priority Registers for Memory Management, Bus Fault and Usage Fault. Unused bits return "0" when read, and writing to unused bits has no effect.

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

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

7.6.2.13 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: Enabled
17	BUSFAUL TENA	R/W	Bus Fault 0: Disable 1: Enable
16	MEMFAULT ENA	R/W	Memory Management 0: Disable 1: Enable
15	SVCALL PENDED	R/W	SVCall 0: Not pended 1: Pended
14	BUSFAULT PENDED	R/W	Bus Fault 0: Not pended 1: Pended
13	MEMFAULT PENDED	R/W	Memory Management 0: Not pended 1: Pended
12	USGFAULT PENDED	R/W	Usage Fault 0: Not pended 1: Pended
11	SYSTICKACT	R/W	SysTick 0: Inactive 1: Active
10	PENDSVACT	R/W	PendSV 0: Inactive 1: Active
9	-	R	Read as 0,
8	MONITORACT	R/W	Debug monitor 0: Inactive 1: Active
7	SVCALLACT	R/W	SVCall 0: Inactive 1: Active
6-4	-	R	Read as 0,

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

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

7.6.3 Clock generator registers

7.6.3.1 CGIMCGA (CG Interrupt Mode Control Register A)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	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
31-23	−	R	Read as 0.
22-20	EMCG2[2:0]	R/W	active level setting of INTC standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
19-18	EMST2[1:0]	R	active level of INTC standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edge
17	−	R	Reads as undefined.
16	INT2EN	R/W	INTC clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCG1[2:0]	R/W	active level setting of INT7 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
11-10	EMST1[1:0]	R	active level of INT7 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edge
9	−	R	Reads as undefined.
8	INT1EN	R/W	INT7 clear input 0: Disable 1: Enable
7	−	R	Read as 0,

Bit	Bit Symbol	Type	Function
6-4	EMCG0[2:0]	R/W	active level setting of INT6 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
3-2	EMST0[1:0]	R	active level of INT6 standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edge
1	−	R	Reads as undefined.
0	INT0EN	R/W	INT6 clear input 0: Disable 1: Enable

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

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

7.6.3.2 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: INT6 0_1000: Reserved 0_0001: INT7 0_1001: Reserved 0_0010: INTC 0_1010: Reserved 0_0011: Reserved 0_1011: Reserved 0_0100: Reserved 0_1100: Reserved 0_0101: Reserved 0_1101: Reserved 0_0110: Reserved 0_1110: Reserved 0_0111: Reserved 0_1111: Reserved 0_0011 to 1_1111: Reserved Read as 0.

7.6.3.3 CGNMIFLG (NMI Flag Register)

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

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

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

7.6.3.4 CGRSTFLG (Reset Flag Register)

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

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

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

Note 2: This product has power-on reset circuit and this register is initialized only by power-on reset. Therefore, "1" is set to the <PONRSTF> bit in initial reset state right after power-on. Note that this bit is not set by the second and subsequent resets and this register is not cleared automatically. Write "0" to clear the register.

8. Digital Noise Filter Circuit (DNF)

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

8.1 Configuration

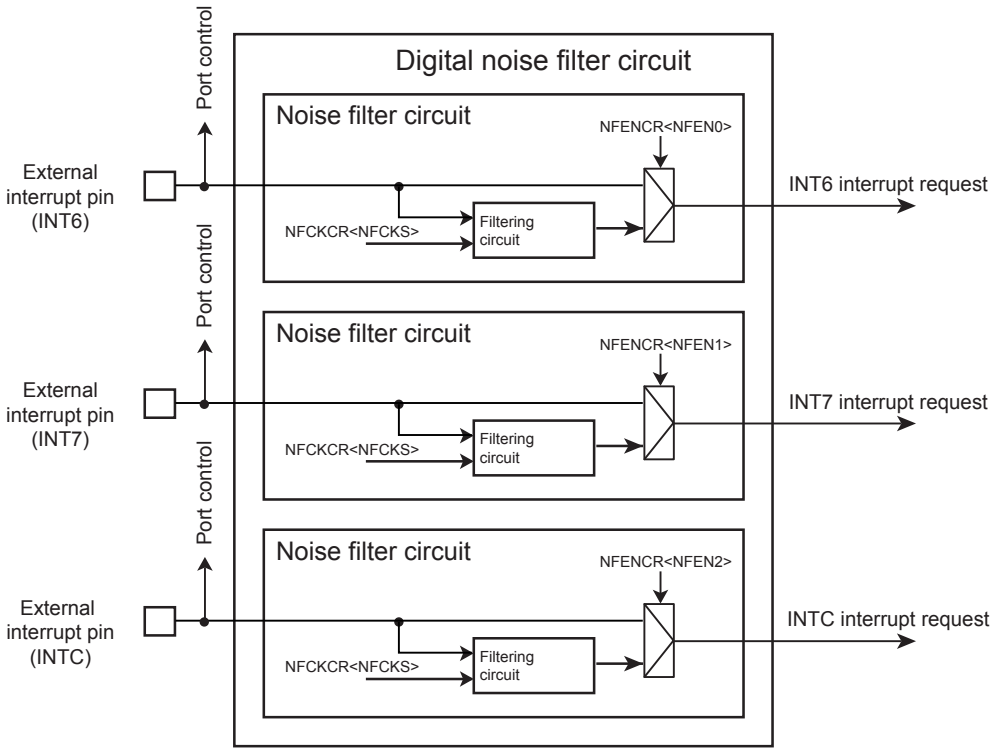


Figure 8-1 Circuit diagram of digital noise filter

8.2 Registers

8.2.1 Register List

Base Address = 0x4006_0000

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

8.2.1.1 NFCKCR (Noise Filter Control Register)

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

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

Note1 : NFCKCR<NFCKS> setting is specified in NFENCR<NFEN[2:0]>="000".

Note2 : If external inputs are used to release STOP mode, the noise filter circuit cannot be used. Make sure to disable the noise filter enable bit of NFENCR register and stop the clock by NFCKCR register.

8.2.1.2 NFENCR (Noise Filter Enable register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	NFEN2	NFEN1	NFEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2	NFENC	R/W	INTC noise filter is enabled. 0: Disabled (Pre-noise filtering output signal and noise filter circuit counter are cleared when releasing STOP mode.) 1: Enabled (Post-noise filtering output signal)
1	NFEN7	R/W	INT7 noise filter is enabled. 0: Disabled (Pre-noise filtering output signal and noise filter circuit counter are cleared when releasing STOP mode.) 1: Enabled (Post-noise filtering output signal)
0	NFEN6	R/W	INT6 noise filter is enabled. 0: Disabled (Pre-noise filtering output signal and noise filter circuit counter are cleared when releasing STOP mode.) 1: Enabled (Post-noise filtering output signal)

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

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

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

8.3 Operation Description

8.3.1 Configuration

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

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

8.3.2 Operation

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

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

8.3.3 Noise Filter Usable Operation Mode

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

8.3.4 Precautions on Use of STOP Mode

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

8.3.5 Minimum Noise Filtering Time

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

Table 8-1 Minimum noise filtering time

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

9. Input / Output Ports

9.1 Port Functions

9.1.1 Function list

TMPM3U0FSDMG has 21 ports. Besides the ports function, these ports can be used as I/O pins for periph-eral functions.

Table 9-1 shows the port function table.

Table 9-1 Port Function List

Port	PIn	Input /Out-put	Pull-up Pull-down	Schmitt Input	Digital Noise Fil-ter circuit	Program-mable Open-drain	Function pin
PORTB							
	PB3	I/O	Pull-up / Pull-down	o	-	o	TMS / SWDIO , (RXD1)
	PB4	I/O	Pull-up / Pull-down	o	-	o	TCK / SWCLK , (TXD1)
	PB5	I/O	Pull-up / Pull-down	o	-	o	TDO / SWV , SCK0, (SDA0 / SO0)
	PB6	I/O	Pull-up / Pull-down	o	o	o	TDI , SCL0 / SI0 , TB7OUT , INT6, RXD1
PORTE							
	PE0	I/O	Pull-up / Pull-down	o	-	o	TXD0 , ENCA
	PE1	I/O	Pull-up / Pull-down	o	-	o	RXD0 , TB4IN, ENCB
	PE2	I/O	Pull-up / Pull-down	o	o	o	SCLK0 , $\overline{\text{CTS0}}$, ENCZ , INT7, (SCL0)
PORTF							
	PF0	I/O	Pull-up / Pull-down	o	o	o	TB7IN , SDA0 / SO0 , $\overline{\text{BOOT}}$, INTC, TXD1
PORTG							
	PG0	I/O	Pull-up / Pull-down	o	-	o	U0
	PG1	I/O	Pull-up / Pull-down	o	-	o	X0
	PG2	I/O	Pull-up / Pull-down	o	-	o	V0
	PG3	I/O	Pull-up / Pull-down	o	-	o	Y0
	PG4	I/O	Pull-up / Pull-down	o	-	o	W0
	PG5	I/O	Pull-up / Pull-down	o	-	o	Z0
	PG6	I/O	Pull-up / Pull-down	o	-	o	$\overline{\text{EMG}}$, $\overline{\text{OVV}}$
PORTJ							
	PJ6	I/O	Pull-up / Pull-down	o	-	o	AINB9
	PJ7	I/O	Pull-up / Pull-down	o	-	o	AINB10
PORTK							
	PK0	I/O	Pull-up / Pull-down	o	-	o	AINB11
	PK1	I/O	Pull-up / Pull-down	o	-	o	AINB12
PORTM							
	PM0	I/O	Pull-up / Pull-down	o	-	o	X1
	PM1	I/O	Pull-up / Pull-down	o	-	o	X2

o : Exist

- : Not Exist

9.1.2 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 function.

  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 programmable 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.

9.1.3 Port States in STOP Mode

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

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

Table 9-2 shows the pin conditions in STOP mode.

Table 9-2 Port conditions in STOP mode

	Pin name	I/O	<DRVE> = 0	<DRVE> = 1
Not port	$\overline{\text{RESET}}$, MODE	Input only	o	
	VOUT15, VOUT3	Output only	o	
Port	X1	Input only	×	
	X2	Output only	"High" level output	
	TMS TCK TDI	Input	o	
	TDO/SWV	Output	Enabled when data is valid. Disabled when data is invalid.	
	SWCLK	Input	o	
	SWDIO	Input	o	
		Output	Enabled when data is valid. Disabled when data is invalid.	
	U0 V0 W0 X0 Y0 Z0	Output	Enabled when data is valid. Disabled when data is invalid.	
	INT6, INT7, INTC	Input	o	
	Other function pins other than the above or the ports that are used as general purpose ports.	Input	×	o
		Output	×	o

o : Input or output enabled.

× : Input or output disabled.

9.2 Port functions

This chapter describes the port registers detail.

This chapter describes only "circuit type" reading circuit configuration. For detailed circuit diagram, refer to the chapter of "Port Section Equivalent Circuit Schematic".

9.2.1 Port B (PB3 to PB6)

The port B is a general-purpose, 4-bit input / output port.For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port B performs the debug interface function, the serial interface function (SIO/UART), the external signal interrupt input and the 16-bit timer output function.

Reset initializes PB3, PB4, PB5, PB6 to perform debug interface function.

When PB3 functions as the TMS or SWDIO, input, output and pull-up are enabled. When PB4 functions as the TCK or SWCLK, input, pull-down are enabled.

When PB5 functions as the TDO or SWV, output is enabled. When PB6 functions TDI, input, pull-up are enabled.

To use the external interrupt input for releasing STOP mode, select this function in the PBFR4 and enable input in the PBIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock / mode control block is set to stop driving of pins during STOP mode.

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

9.2.1.1 Port B register

Base Address = 0x4000_0040

Register name		Address(Base+)
Port B data register	PBDATA	0x0000
Port B output control register	PBCR	0x0004
Port B function register 1	PBFR1	0x0008
Port B function register 2	PBFR2	0x000C
Port B function register 3	PBFR3	0x0010
Port B function register 4	PBFR4	0x0014
Port B function register 5	PBFR5	0x0018
Port B open drain control register	PBOD	0x0028
Port B pull-up control register	PBPUP	0x002C
Port B pull-down control register	PBPDN	0x0030
Port B input control register	PBIE	0x0038

9.2.1.2 PBDATA (Port B data register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-3	PB6 to PB3	R/W	Port B data register
2-0	−	R	Read as 0.

9.2.1.3 PBCR (Port B output control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-3	PB6C to PB3C	R/W	Output 0: Disable 1: Enable
2-0	−	R	Read as 0.

9.2.1.4 PBFR1 (Port B function register 1)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PB6F1	R/W	0: PORT 1: TDI
5	PB5F1	R/W	0: PORT 1: TDO / SWV
4	PB4F1	R/W	0: PORT 1: TCK / SWCLK
3	PB3F1	R/W	0: PORT 1: TMS / SWDIO
2-0	−	R	Read as 0.

9.2.1.5 PBFR2 (Port B function register 2)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PB6F2	R/W	0: PORT 1: SCL0 / SI0 (Note)
5-0	−	R	Read as 0.

Note: Both PB6 pin and PE2 pin are assingned the SCL0 function. When this function is used, it should be set as follows.

SCL0	<PB6F2>	<PE2F5>
PB6	1	0
PE2	0	1
If <PB6F2> and <PE2F5> are set to "1", the PB6 is valid for this function.		

9.2.1.6 PBFR3 (Port B function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6F3	PB5F3	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PB6F3	R/W	0: PORT 1: TB7OUT
5	PB5F3	R/W	0: PORT 1: SCK0
4-0	−	R	Read as 0.

9.2.1.7 PBFR4 (Port B function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6F4	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PB6F4	R/W	0: PORT 1: INT6
5-0	−	R	Read as 0.

9.2.1.8 PBFR5 (Port B function register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PB6F5	PB5F5	PB4F5	PB3F5	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PB6F5	R/W	0: PORT 1: RXD1 (Note)
5	PB5F5	R/W	0: PORT 1: SDA0 / SO0 (Note)
4	PB4F5	R/W	0: PORT 1: TXD1 (Note)
3	PB3F5	R/W	0: PORT 1: RXD1 (Note)
2-0	−	R	Read as 0.

Note: Both PB6 pin and PB3 pin are assingned the RXD1 function, both PB5 pin and PF0 pin are assingned the SDA0/SO0 function, moreover, both PB4 pin and PF0 pin are assingned the TXD1 function as well. When these function are used, it should be set as follows.

RXD1	<PB6F5>	<PB3F5>
PB6	1	0
PB3	0	1
If <PB6F5> and <PB3F5> are set to "1", the PB6 is valid for this function.		

SDA0/SO0	<PF0F2>	<PB5F5>
PF0	1	0
PB5	0	1
If <PF0F2> and <PB5F5> are set to "1", the PF0 is valid for this function.		

TXD1	<PF0F5>	<PB4F5>
PF0	1	0
PB4	0	1
If <PF0F5> and <PB4F5> are set to "1", the PF0 is valid for this function.		

9.2.1.9 PBOD (Port B open drain control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-3	PB6OD to PB3OD	R/W	0 : CMOS 1 : Open-drain
2-0	−	R	Read as 0.

9.2.1.10 PBPUP (Port B pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-3	PB6UP to PB3UP	R/W	Pull-up 0: Disable 1: Enable
2-0	−	R	Read as 0.

9.2.1.11 PBPDN (Port B pull-down control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-3	PB6DN to PB3DN	R/W	Pull-down 0: Disable 1: Enable
2-0	−	R	Read as 0.

9.2.1.12 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	-	PB6IE	PB5IE	PB4IE	PB3IE	-	-	-
After reset	0	1	0	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-3	PB6IE to PB3IE	R/W	Input 0: Disable 1: Enable
2-0	−	R	Read as 0.

9.2.2 Port E (PE0 to PE2)

The port E is a general-purpose, 3-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port E performs the serial interface function (SIO / UART), the serial bus interface function (I2C / SIO), the Encoder input function, the external signal interrupt input and the 16-bit timer input function.

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

The Port E have several types of function register. If you use the port E as a general-purpose port, set "0" to the corresponding bit of the several 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 some function registers at the same time.

To use the external interrupt input for releasing STOP mode, select this function in the PEFR4 and enable input in the PEIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock / mode control block is set to stop driving of pins during STOP mode.

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

9.2.2.1 Port E register

Base Address = 0x4000_0100

Register name		Address(Base+)
Port E data register	PEDATA	0x0000
Port E output control register	PECR	0x0004
Port E function register 1	PEFR1	0x0008
Port E function register 2	PEFR2	0x000C
Port E function register 3	PEFR3	0x0010
Port E function register 4	PEFR4	0x0014
Port E function register 5	PEFR5	0x0018
Port E open drain control register	PEOD	0x0028
Port E pull-up control register	PEPUP	0x002C
Port E pull-down control register	PEPDN	0x0030
Port E input control register	PEIE	0x0038

9.2.2.2 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	-	-	-	-	-	PE2	PE1	PE0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PE2 to PE0	R/W	Port E data register

9.2.2.3 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	-	-	-	-	-	PE2C	PE1C	PE0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PE2C to PE0C	R/W	Output 0: Disable 1: Enable

9.2.2.4 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	-	-	-	-	-	PE2F1	PE1F1	PE0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PE2F1	R/W	0: PORT 1: SCLK0
1	PE1F1	R/W	0: PORT 1: RXD0
0	PE0F1	R/W	0: PORT 1: TXD0

9.2.2.5 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	-	-	-	-	-	PE2F2	PE1F2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PE2F2	R/W	0: PORT 1: $\overline{\text{CTS0}}$
1	PE1F2	R/W	0: PORT 1: TB4IN
0	−	R	Read as 0.

9.2.2.6 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	-	-	-	-	-	PE2F3	PE1F3	PE0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PE2F3	R/W	0: PORT 1: ENCZ
1	PE1F3	R/W	0: PORT 1: ENCB
0	PE0F3	R/W	0: PORT 1: ENCA

9.2.2.7 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: INT7
1-0	−	R	Read as 0.

9.2.2.8 PEFR5 (Port E function register 5)

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

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PE2F5	R/W	0: PORT 1: SCL0 / SI0 (Note)
1-0	−	R	Read as 0.

Note: Both PB6 pin and PE2 pin are assingned the SCL0 function. When this function is used, it should be set as follows.

SCL0	<PB6F2>	<PE2F5>
PB6	1	0
PE2	0	1
If <PB6F2> and <PE2F5> are set to "1", the PB6 is valid for this function.		

9.2.2.9 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	-	-	-	-	-	PE2OD	PE1OD	PE0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PE2OD to PE0OD	R/W	0 : CMOS 1 : Open-drain

9.2.2.10 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	-	-	-	-	-	PE2UP	PE1UP	PE0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PE2UP to PE0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.2.11 PEPDN (Port E pull-down control register)

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

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PE2DN to PE0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.2.12 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	-	-	-	-	-	PE2IE	PE1IE	PE0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2-0	PE2IE to PE0IE	R/W	Input 0: Disable 1: Enable

9.2.3 Port F (PF0)

The port F is a general-purpose, 1-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 F performs the serial bus interface function (I2C / SIO), the serial interface function (UART), the 16-bit timer input function and the operation mode setting.

While "0" is input the reset input pin and power-on reset period, the PF0 input and pull-up are enabled. At the rising edge of the reset signal, if PF0 is "1", the device enters single mode and boots from the on-chip flash memory. If PF0 is "0", the device enters single boot mode and boots from the internal boot program. For details of single boot mode, refer to Chapter "Flash Memory Operation".

Reset initializes a bit of the port F function registers as general-purpose ports with input, output, pull-up and pull-down disabled.

After reset sequence, the PF0 is disabled with input and pull-up based on initial setting of PFIE and PFPUP registers.

To use the external interrupt input for releasing STOP mode, select this function in the PFFR4 and enable input in the PFIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock / mode control block is set to stop driving of pins during STOP mode.

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

9.2.3.1 Port F register

Base Address = 0x4000_0140

Register name		Address(Base+)
Port F data register	PFDATA	0x0000
Port F output control register	PFCR	0x0004
Port F function register 1	PFFR1	0x0008
Port F function register 2	PFFR2	0x000C
Port F function register 4	PFFR4	0x0014
Port F function register 5	PFFR5	0x0018
Port F open drain control register	PFOD	0x0028
Port F pull-up control register	PFPUP	0x002C
Port F pull-down control register	PFPDN	0x0030
Port F input control register	PFIE	0x0038

9.2.3.2 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	-	-	-	-	-	-	-	PF0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	PF0	R/W	Port F data register

9.2.3.3 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	-	-	-	-	-	-	-	PF0C
After reset	0	0	0	0	0	0	0	0

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

9.2.3.4 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	-	-	-	-	-	-	-	PF0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	PF0F1	R/W	0: PORT 1: TB7IN

9.2.3.5 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	-	-	-	-	-	-	-	PF0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	PF0F2	R/W	0: PORT 1: SDA0/SO0 (Note)

Note: Both PB5 pin and PF0 pin are assingned the SDA0/SO0 function. When this function is used, it should be set as follows.

SDA0/SO0	<PF0F2>	<PB5F5>
PF0	1	0
PB5	0	1
If <PF0F2> and <PB5F5> are set to "1", the PF0 is valid for this function.		

9.2.3.6 PFFR4 (Port F function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	PF0F4
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	PF0F4	R/W	0: PORT 1: INTC

9.2.3.7 PFFR5 (Port F function register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	PF0F5
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	PF0F5	R/W	0: PORT 1: TXD1 (Note)

Note: Both PB4 pin and PF0 pin are assingned the TXD1 function. When this function is used, it should be set as follows.

TXD1	<PF0F5>	<PB4F5>
PF0	1	0
PB4	0	1
If <PF0F5> and <PB4F5> are set to "1", the PF0 is valid for this function.		

9.2.3.8 PFOD (Port F open drain control register)

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

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

9.2.3.9 PFPUP (Port F pull-up control register)

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

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

9.2.3.10 PFPDN (Port F pull-down control register)

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

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

9.2.3.11 PFIE (Port F input control register)

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

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

9.2.4 Port G (PG0 to PG6)

The port G 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 G performs the input/output port for three-phase motor control (PMD) function.

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

9.2.4.1 Port G register

Base Address = 0x4000_0180

Register name		Address(Base+)
Port G data register	PGDATA	0x0000
Port G output control register	PGCR	0x0004
Port G function register 1	PGFR1	0x0008
Port G function register 2	PGFR2	0x000C
Port G open drain control register	PGOD	0x0028
Port G pull-up control register	PGPUP	0x002C
Port G pull-down control register	PGPDN	0x0030
Port G input control register	PGIE	0x0038

9.2.4.2 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	-	PG6	PG5	PG4	PG3	PG2	PG1	PG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PG6 to PG0	R/W	Port G data register

9.2.4.3 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	-	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PG6C to PG0C	R/W	Output 0: Disable 1: Enable

9.2.4.4 PGFR1 (Port G function register 1)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PG6F1	R/W	0: PORT 1: $\overline{\text{EMG}}$
5	PG5F1	R/W	0: PORT 1: Z0
4	PG4F1	R/W	0: PORT 1: W0
3	PG3F1	R/W	0: PORT 1: Y0
2	PG2F1	R/W	0: PORT 1: V0
1	PG1F1	R/W	0: PORT 1: X0
0	PG0F1	R/W	0: PORT 1: U0

9.2.4.5 PGFR2 (Port G function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PG6F2	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PG6F1	R/W	0: PORT 1: $\overline{OVV}$
5-0	−	R	Read as 0.

9.2.4.6 PGOD (Port G open drain control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PG6OD to PG0OD	R/W	0 : CMOS 1 : Open-drain

9.2.4.7 PGPUP (Port G pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PG6UP to PG0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.4.8 PGPDN (Port G pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PG6DN	PG5DN	PG4DN	PG3DN	PG2DN	PG1DN	PG0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PG6DN to PG0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.4.9 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	-	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PG6IE to PG0IE	R/W	Input 0: Disable 1: Enable

9.2.5 Port J (PJ6 to PJ7)

The port J is a general-purpose, 2-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 analog input of the AD converter.

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

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

9.2.5.1 Port J register

Base Address = 0x4000_0240

Register name		Address(Base+)
Port J data register	PJDATA	0x0000
Port J output control register	PJCR	0x0004
Port J open drain control register	PJOD	0x0028
Port J pull-up control register	PJPUP	0x002C
Port J pull-down control register	PJPDN	0x0030
Port J input control register	PJIE	0x0038

9.2.5.2 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	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	PJ7 to PJ6	R/W	Port J data register
5-0	−	R	Read as 0.

9.2.5.3 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	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	PJ7C to PJ6C	R/W	Output 0: Disable 1: Enable
5-0	−	R	Read as 0.

9.2.5.4 PJOD (Port J 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	PJ7OD	PJ6OD	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	PJ7OD to PJ6OD	R/W	0 : CMOS 1 : Open-drain
5-0	−	R	Read as 0.

9.2.5.5 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	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	PJ7UP to PJ6UP	R/W	Pull-up 0: Disable 1: Enable
5-0	−	R	Read as 0.

9.2.5.6 PJPDN (Port J 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	PJ7DN	PJ6DN	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	PJ7DN-PJ6DN	R/W	Pull-down 0: Disable 1: Enable
5-0	−	R	Read as 0.

9.2.5.7 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	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-6	PJ7IE to PJ6IE	R/W	Input 0: Disable 1: Enable
5-0	−	R	Read as 0.

9.2.6 Port K (PK0 to PK1)

The port K is a general-purpose, 2-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 analog input of the AD converter.

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

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

9.2.6.1 Port K register

Base Address = 0x4000_0280

Register name		Address(Base+)
Port K data register	PKDATA	0x0000
Port K output control register	PKCR	0x0004
Port K open drain control register	PKOD	0x0028
Port K pull-up control register	PKPUP	0x002C
Port K pull-down control register	PKPDN	0x0030
Port K input control register	PKIE	0x0038

9.2.6.2 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	-	-	-	-	-	-	PK1	PK0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PK1 to PK0	R/W	Port K data register

9.2.6.3 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	-	-	-				PK1C	PK0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PK1C to PK0C	R/W	Output 0: Disable 1: Enable

9.2.6.4 PKOD (Port K open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1OD	PK0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PK1OD to PK0OD	R/W	0 : CMOS 1 : Open-drain

9.2.6.5 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	-	-	-	-	-	-	PK1UP	PK0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PK1UP to PK0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.6.6 PKPDN (Port K pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1DN	PK0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PK1DN-PK0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.6.7 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	-	-	-	-	-	-	PK1IE	PK0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PK1IE-PK0IE	R/W	Input 0: Disable 1: Enable

9.2.7 Port M (PM0 to PM1)

The port M is a general-purpose, 2-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 M performs the high-speed oscillator1(X1 and X2) by CGOSCCR<HOSCON>=1.

While it become CGOSCCR<HOSCON>=1, each register of port M can not change to write. The procedure when it is used as an outside high-speed oscillator connection terminal, refer to a chapter of the "system clock".(Note1)

Reset initializes all bits of the port M as general-purpose ports with input, output, pull-up and pull-down disabled.(Note2)

Note 1: If one of the Port M registers except PMDATA and PMOD is not equal to the initial value, CGOSCCR<HOSCON> can not be set to "1".

Note 2: The high-speed clock chosen after reset cancellation is a built-in high-speed clock. Therefore, in the initial state, it become port M.

9.2.7.1 Port M register

Base Address = 0x4000_0300

Register name		Address(Base+)
Port M data register	PMDATA	0x0000
Port M output control register	PMCR	0x0004
Port M open drain control register	PMOD	0x0028
Port M pull-up control register	PMPUP	0x002C
Port M pull-down control register	PMPDN	0x0030
Port M input control register	PMIE	0x0038

9.2.7.2 PMDATA (Port M 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	-	-	-	-	-	-	PM1	PM0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PM1 to PM0	R/W	Port M data register

9.2.7.3 PMCR (Port M 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	-	-	-	-	-	-	PM1C	PM0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PM1C to PM0C	R/W	Output 0: Disable 1: Enable

9.2.7.4 PMOD (Port M 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	-	-	-	-	-	-	PM1OD	PM0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PM1OD to PM0OD	R/W	0 : CMOS 1 : Open-drain

9.2.7.5 PMPUP (Port M 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	-	-	-	-	-	-	PM1UP	PM0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PM1UP to PM0UP	R/W	Pull-up 0: Disable 1: Enable

9.2.7.6 PMPDN (Port M 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	-	-	-	-	-	-	PM1DN	PM0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PM1DN to PM0DN	R/W	Pull-down 0: Disable 1: Enable

9.2.7.7 PMIE (Port M 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	-	-	-	-	-	-	PM1IE	PM0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PM1IE to PM0IE	R/W	Input 0: Disable 1: Enable

9.3 Block Diagrams of Ports

9.3.1 Port Types

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

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

Table 9-3 Function lists

Type	GP Port	Function	Analog	Pull-up	Pull-down	Program-mable open-drain	Note
FT1	I/O	I/O	−	R	−	o	
FT2	I/O	I/O	−	R	−	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	I/O	o	R	−	−	
FT6	Output	Output	−	NoR	−	o	$\overline{\text{BOOT}}$ input enabled during reset

int : Interrupt input
- : Not exist
o : exist
R : Forced disable during reset
NoR : Unaffected by reset

9.3.2 Type FT1

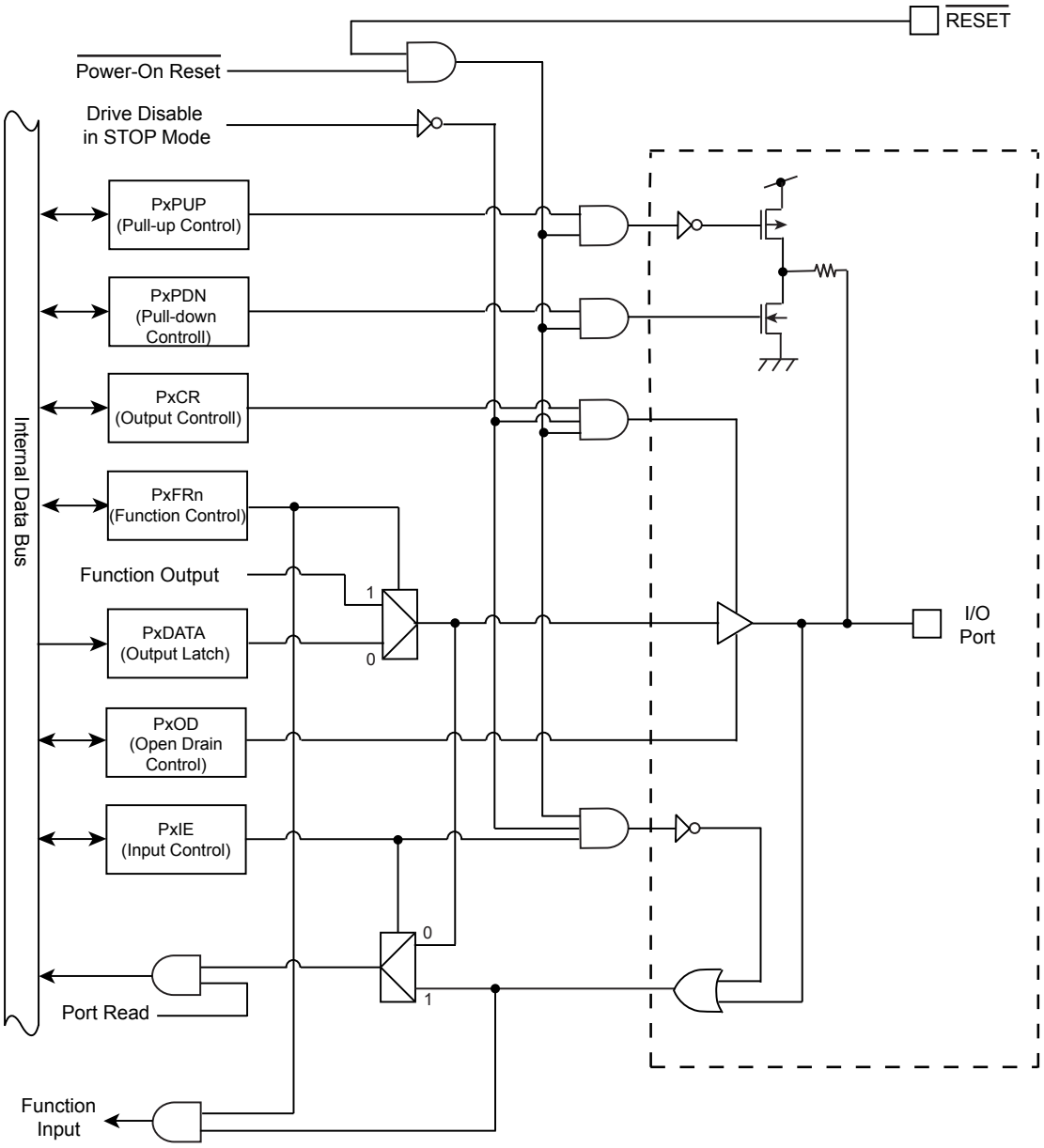


Figure 9-1 Port Type FT1

9.3.3 Type FT2

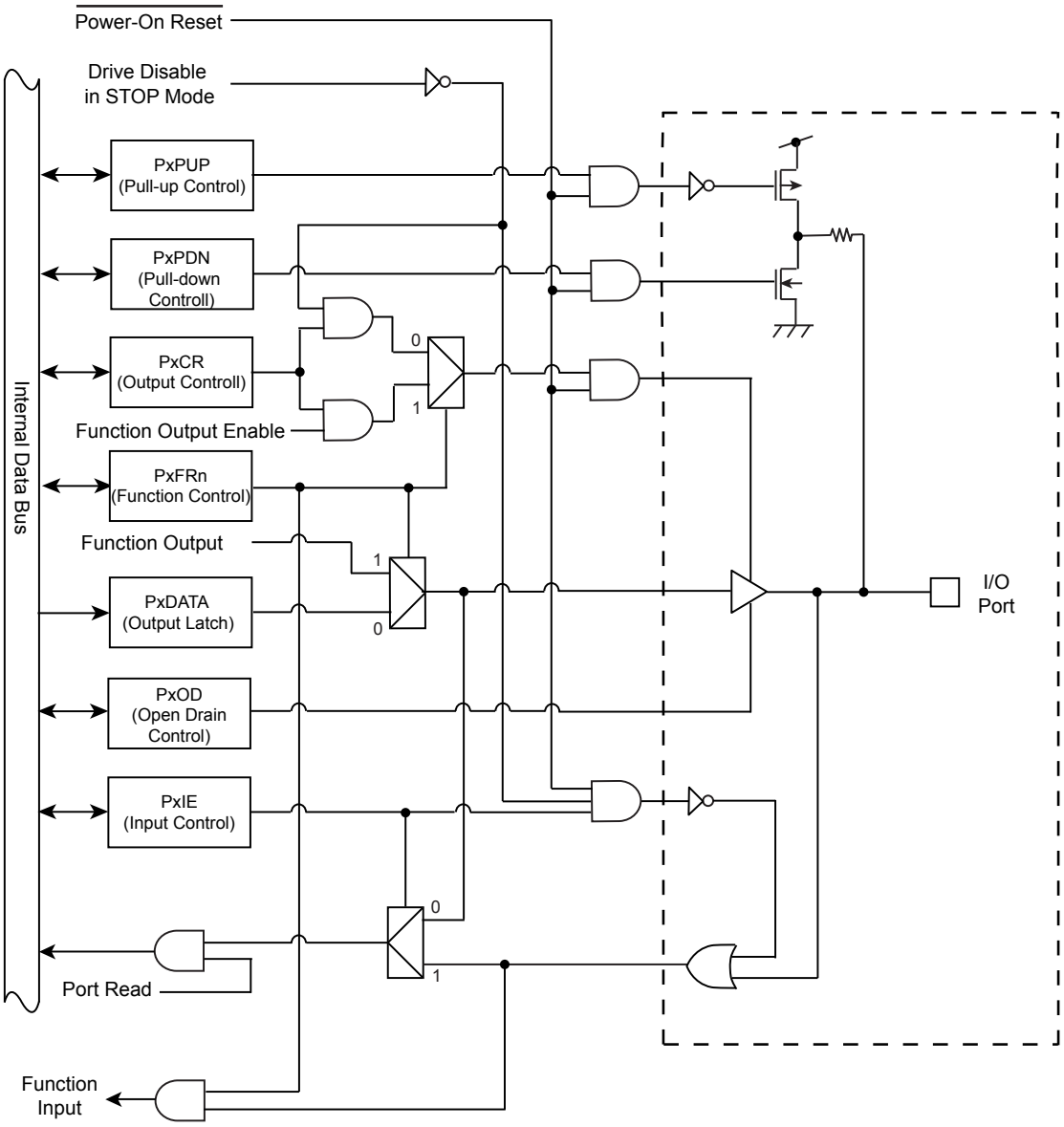


Figure 9-2 Port Type FT2

9.3.4 Type FT3

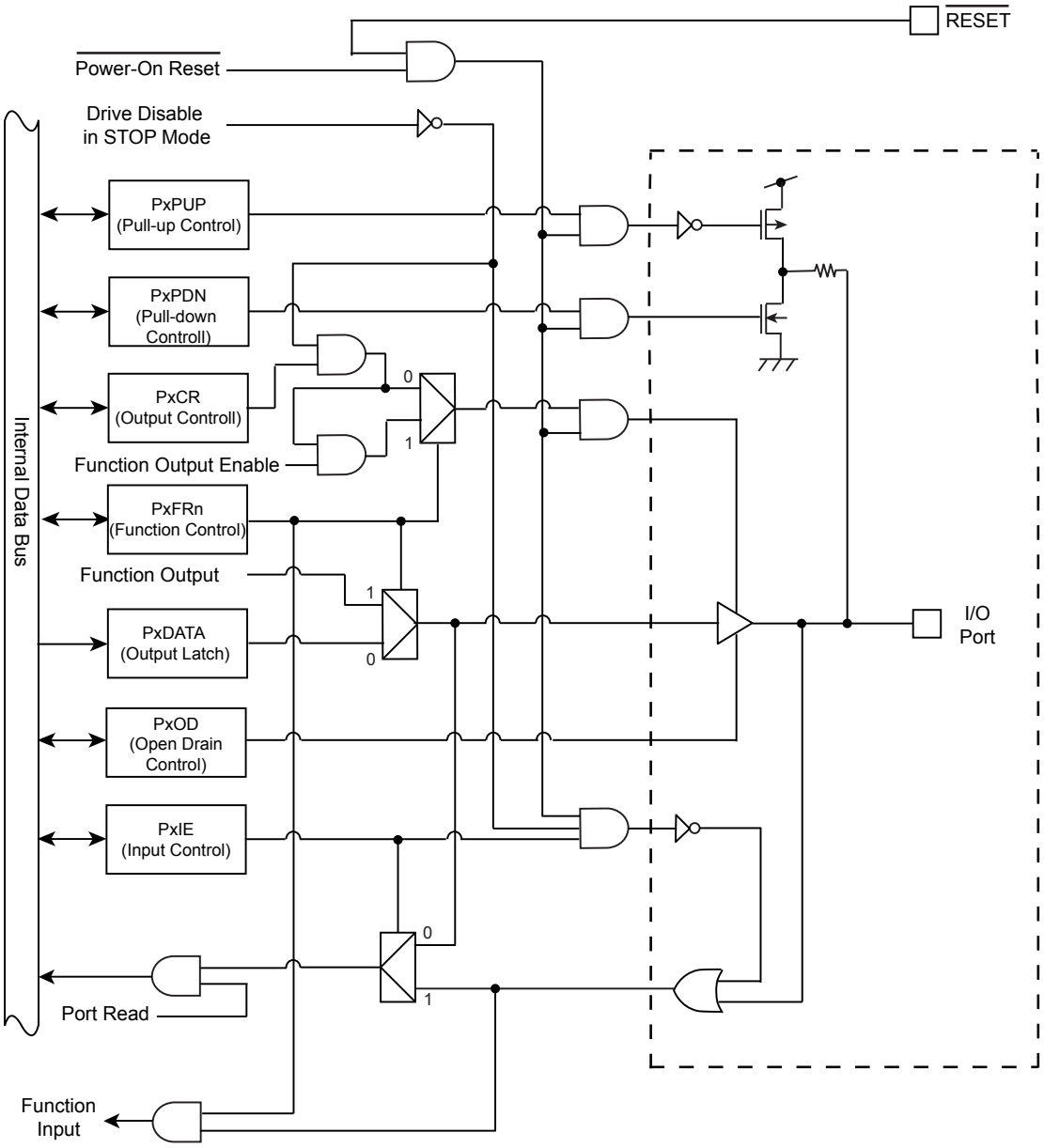


Figure 9-3 Port Type FT3

9.3.5 Type FT4

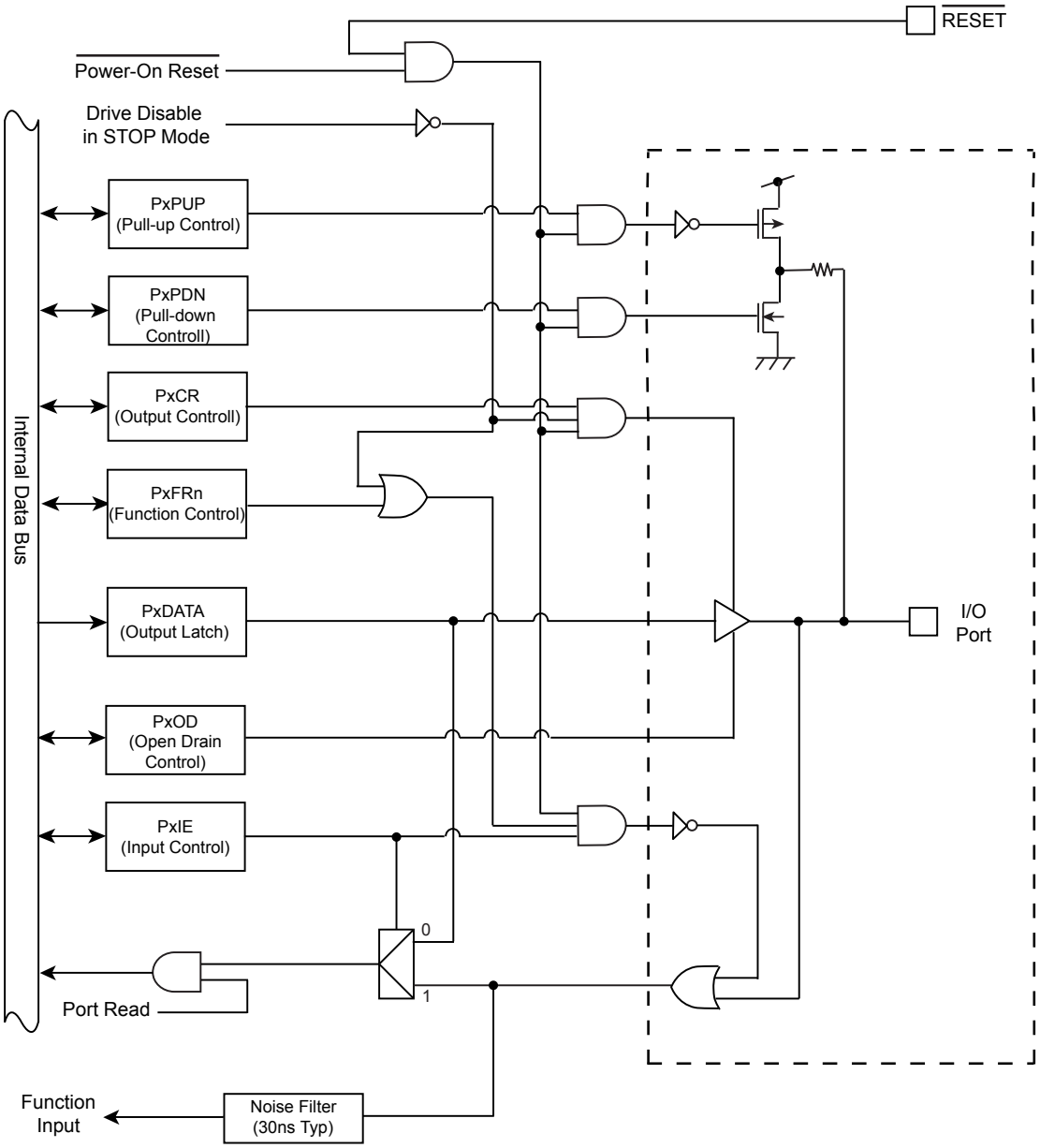


Figure 9-4 Port Type FT4

9.3.6 Type FT5

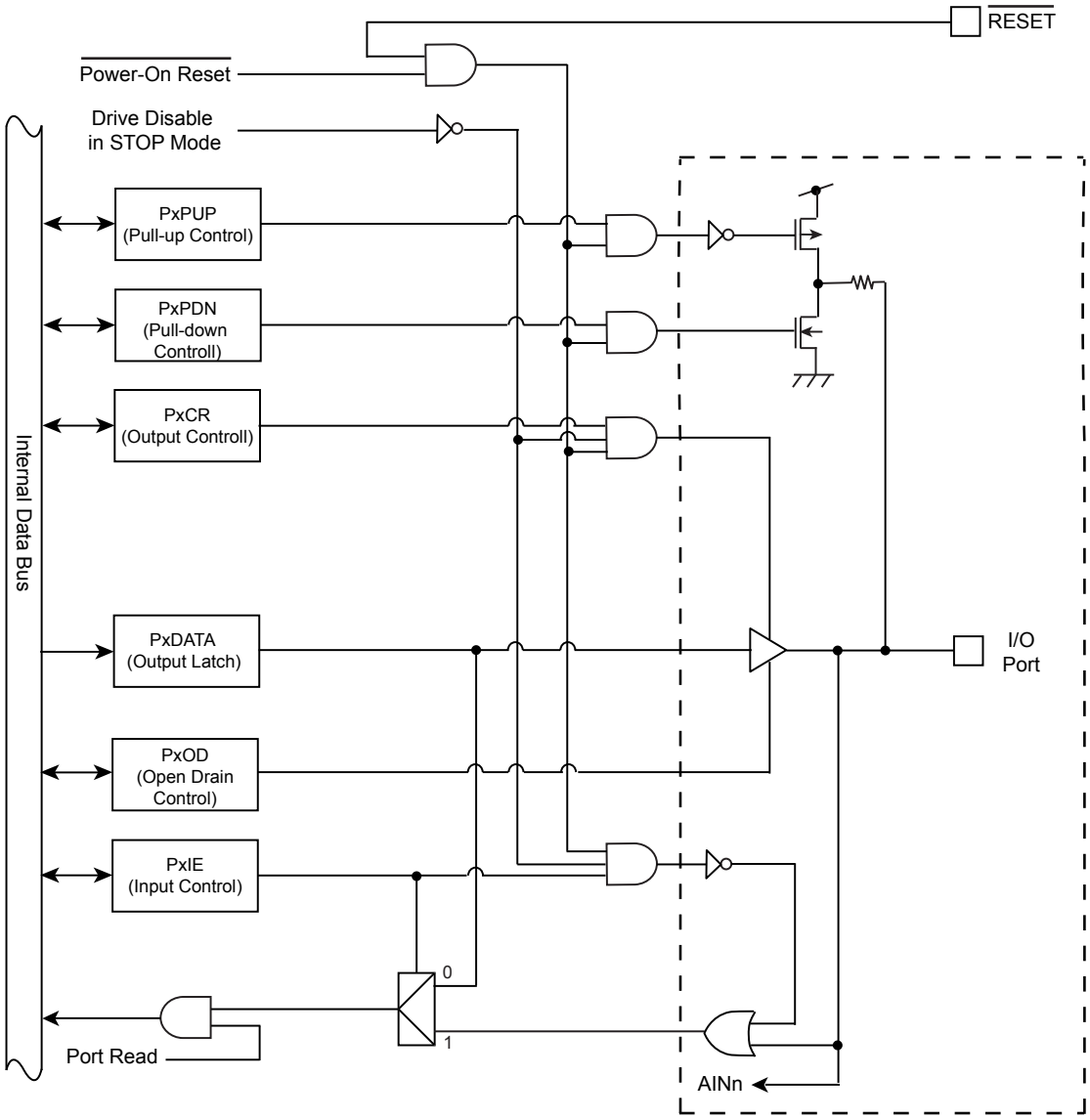


Figure 9-5 Port Type FT5

9.3.7 Type FT6

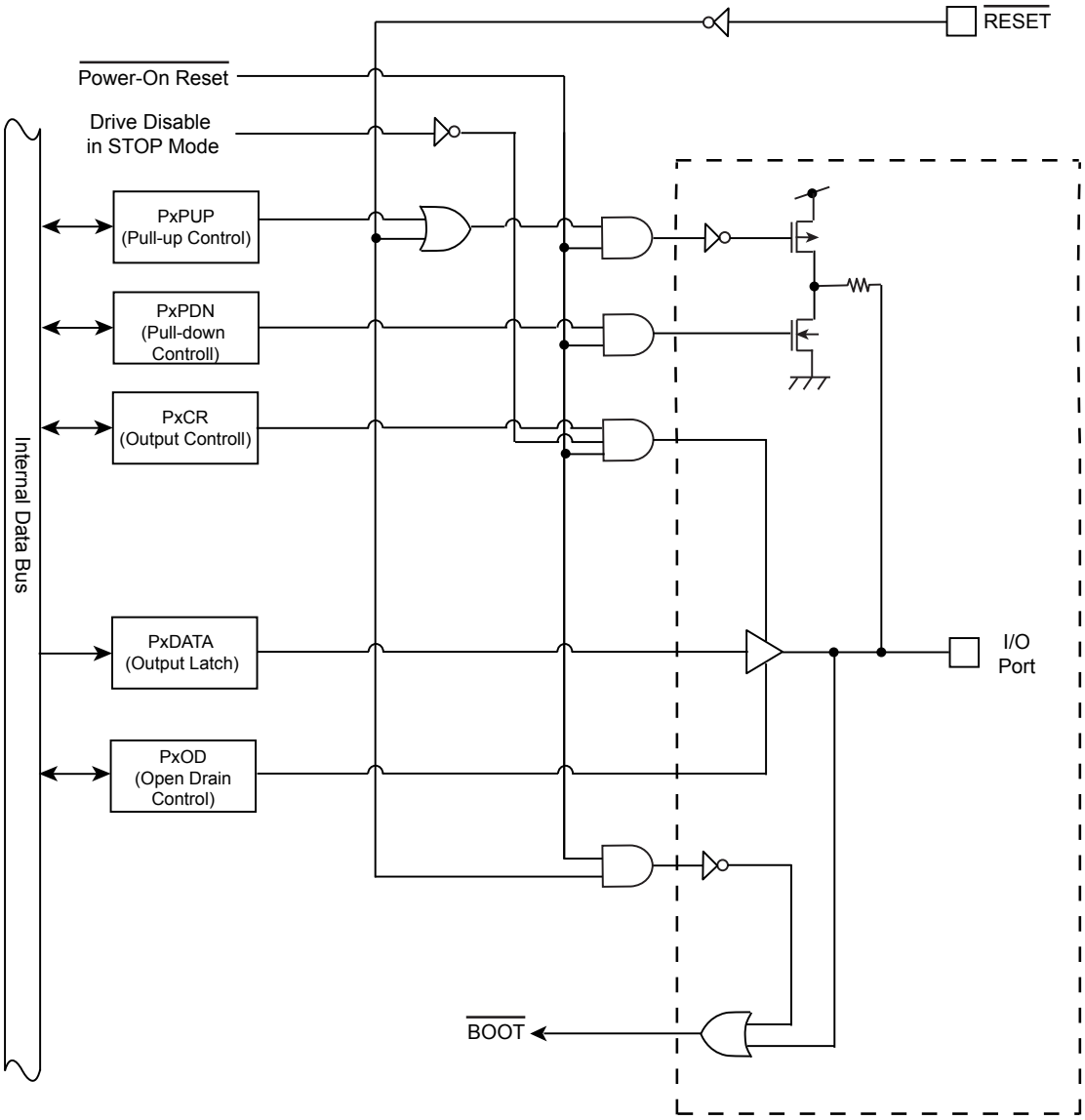


Figure 9-6 Port Type FT6

9.4 Appendix Port Setting List

The following table shows the register setting for each function.

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

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

9.4.1 Port B Setting

Table 9-4 Port Setting List(Port B)

Pin	Port Type	Function	After reset	PB CR	PB FR1	PB FR2	PB FR3	PB FR4	PB FR5	PB OD	PB PUP	PB PDN	PB IE
PB3	-	Input Port		0	0	-	-	-	0	x	x	x	1
		Output Port		1	0	-	-	-	0	x	x	x	0
	FT2	TMS / SWDIO (I / O)	・	1	1	-	-	-	0	0	1	0	1
	FT1	RXD1 (Input)		0	0	-	-	-	1	x	x	x	1
PB4	-	Input Port		0	0	-	-	-	0	x	x	x	1
		Output Port		1	0	-	-	-	0	x	x	x	0
	FT2	TCK / SWCLK (Input)	・	0	1	-	-	-	0	0	0	1	1
	FT1	TXD1 (Output)		1	0	-	-	-	1	x	x	x	0
PB5	-	Input Port		0	0	-	0	-	0	x	x	x	1
		Output Port		1	0	-	0	-	0	x	x	x	0
	FT2	TDO / SWV (Output)	・	1	1	-	0	-	0	0	0	0	0
	FT1	SCK (I / O)		1	0	-	1	-	0	x	x	x	1
		SO0 (Output)		1	0	-	0	-	1	x	x	x	0
		SDA0 (I / O)		1	0	-	0	-	1	x	x	x	1
PB6	-	Input Port		0	0	0	0	0	0	x	x	x	1
		Output Port		1	0	0	0	0	0	x	x	x	0
	FT2	TDI (Input)	・	0	1	0	0	0	0	0	1	0	1
	FT1	SI0 (Input)		0	0	1	0	0	0	x	x	x	1
		SCL0 (I / O)		1	0	1	0	0	0	x	x	x	1
		TB7OUT (Output)		1	0	0	1	0	0	x	x	x	0
	FT4	INT6 (Input)		0	0	0	0	1	0	x	x	x	1
	FT1	RXD1 (Input)		0	0	0	0	0	1	x	x	x	1

9.4.2 Port E Setting

Table 9-5 Port Setting List(Port E)

Pin	Port Type	Function	After reset	PE CR	PE FR1	PE FR2	PE FR3	PE FR4	PE FR5	PE OD	PE PUP	PE PDN	PE IE
PE0	-	Input Port		0	0	-	0	-	-	x	x	x	1
		Output Port		1	0	-	0	-	-	x	x	x	0
	FT1	TXD0 (Output)		1	1	-	0	-	-	x	x	x	0
		ENCA (Input)		0	0	-	1	-	-	x	x	x	1
PE1	-	Input Port		0	0	0	0	-	-	x	x	x	1
		Output Port		1	0	0	0	-	-	x	x	x	0
	FT1	RXD0 (Input)		0	1	0	0	-	-	x	x	x	1
		TB4IN (Input)		0	0	1	0	-	-	x	x	x	1
		ENCB (Input)		0	0	0	1	-	-	x	x	x	1
PE2	-	Input Port		0	0	0	0	0	0	x	x	x	1
		Output Port		1	0	0	0	0	0	x	x	x	0
	FT1	SCLK0 (I / O)		1	1	0	0	0	0	x	x	x	1
		$\overline{\text{CTS0}}$ (Input)		0	0	1	0	0	0	x	x	x	1
		ENCZ (Input)		0	0	0	1	0	0	x	x	x	1
	FT4	INT7 (Input)		0	0	0	0	1	0	x	x	x	1
	FT1	SCL0 (I / O)		1	0	0	0	0	1	x	x	x	1

9.4.3 Port F Setting

Table 9-6 Port Setting List(Port F)

Pin	Port Type	Function	After reset	PF CR	PF FR1	PF FR2	PF FR4	PF FR5	PF OD	PF PUP	PF PDN	PF IE
PF0	-	Input Port		0	0	0	0	0	x	x	x	1
		Output Port		1	0	0	0	0	x	x	x	0
	FT1	TB7IN (Input)		0	1	0	0	0	x	x	x	1
		SO0 (Output)		1	0	1	0	0	x	x	x	0
		SDA0 (I / O)		1	0	1	0	0	x	x	x	1
	FT4	INTC (Input)		0	0	0	1	0	x	x	x	1
	FT1	TXD1 (Output)		1	0	0	0	1	x	x	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 in "Low" state. This case, the port type is FT6.

9.4.4 Port G Setting

Table 9-7 Port Setting List(Port G)

Pin	Port Type	Function	After reset	PG CR	PG FR1	PG FR2	PG OD	PG PUP	PG PDN	PG IE
PG0	-	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
	FT2	U0 (Output)		1	1	-	x	x	x	0
PG1	-	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
	FT2	X0 (Output)		1	1	-	x	x	x	0
PG2	-	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
	FT2	V0 (Output)		1	1	-	x	x	x	0
PG3	-	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
	FT2	Y0 (Output)		1	1	-	x	x	x	0
PG4	-	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
	FT2	W0 (Output)		1	1	-	x	x	x	0
PG5	-	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
	FT2	Z0 (Output)		1	1	-	x	x	x	0
PG6	-	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
	FT1	$\overline{\text{EMG}}$ (Input)		0	1	0	x	x	x	1
		$\overline{\text{OVV}}$ (Input)		0	0	1	x	x	x	1

9.4.5 Port J Setting

Table 9-8 Port Setting List(Port J)

Pin	Port Type	Function	After reset	PJ CR	PJ OD	PJ PUP	PJ PDN	PJ IE
PJ6	-	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
	FT5	AINB9 (Input)	・	0	0	0	0	0
PJ7	-	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
	FT5	AINB10 (Input)	・	0	0	0	0	0

9.4.6 Port K Setting

Table 9-9 Port Setting List(Port K)

Pin	Port Type	Function	After reset	PK CR	PK OD	PK PUP	PK PDN	PK IE
PK0	-	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
	FT5	AINB11 (Input)	・	0	0	0	0	0
PK1	-	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
	FT5	AINB12 (Input)	・	0	0	0	0	0

9.4.7 Port M Setting

Table 9-10 Port Setting List(Port M)

Pin	Port Type	Function	After reset	PM CR	PM OD	PM PUP	PM PDN	PM IE
PM0	-	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
PM1	-	Input Port		0	x	x	x	1
		Input Port		1	x	x	x	0

Note:X1 and X2 function are available.

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

10.1 Outline

TMRB operate in the following four operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation mode (PPG)
- External trigger Programmable pulse generation mode (PPG)

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

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

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

10.2 Differences in the Specifications

TMPM3U0FSDMG contains 4-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 10-1.

Table 10-1 Differences in the Specifications of TMRB Modules

Specification / Channel	External pins		Interrupt		Internal connection
	External clock / capture trigger input pins	Timer flip-flop output pin	Capture interrupt	TMRB interrupt	
	Signal name	Signal name			
TMRB0	-	-	INTCAP00 INTCAP01-	INTTB00 INTTB01	TIMPLS -->(Capture input)
TMRB4	TB4IN	-	-	INTTB40 INTTB41	
TMRB5	-	-	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51 -->(ADC conversion start)
TMRB7	TB7IN	TB7OUT	INTCAP70 INTCAP71	INTTB70 INTTB71	

10.3 Configuration

Each channel consists of a 16-bit up-counter, two 16-bit timer registers (double-buffered), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit. Timer operation modes and the timer flip-flop are controlled by a register.

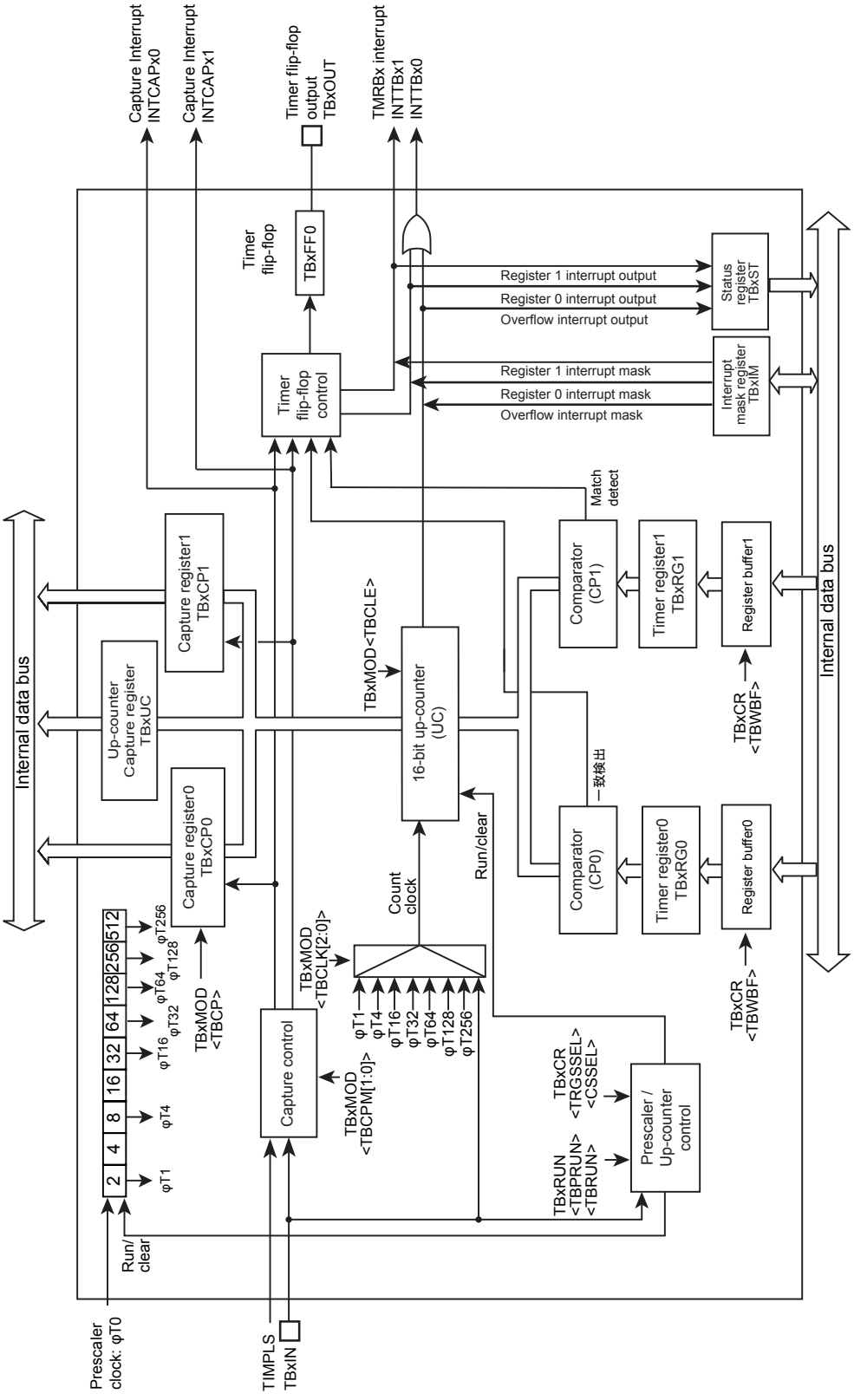


Figure 10-1 TMRBx Block Diagram (x= 0,4,5,7)

10.4 Registers

10.4.1 Register list according to channel

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

Channel x	Base Address
Channel0	0x4001_0000
Channel4	0x4001_0100
Channel5	0x4001_0140
Channel7	0x4001_01C0

Register name (x=0,4,5,7)		Address(Base+)
Enable register	TBxEN	0x0000
RUN register	TBxRUN	0x0004
Control register	TBxCR	0x0008
Mode register	TBxMOD	0x000C
Flip-flop control register	TBxFFCR	0x0010
Status register	TBxST	0x0014
Interrupt mask register	TBxIM	0x0018
Up counter capture register	TBxUC	0x001C
Timer register 0	TBxRG0	0x0020
Timer register 1	TBxRG1	0x0024
Capture register 0	TBxCP0	0x0028
Capture register 1	TBxCP1	0x002C

10.4.2 TBxEN(Enable register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBEN	R/W	TMRBx operation 0: Disable 1: Enable Specifies the TMRB operation. When the operation is disabled, no clock is supplied to the other registers in the TMRB module. This can reduce power consumption. (This disables reading from and writing to the other registers except TBxEN register.) To use the TMRB, enable the TMRB operation (set to "1") before programming each register in the TMRB module. If the TMRB operation is executed and then disabled, the settings will be maintained in each register.
6	TBHALT	R/W	Clock operation during debug HALT. 0: Run 1: Stop
5-0	−	R	Read as "0".

10.4.3 TBxRUN(RUN register)

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

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

Note 1: When the external trigger start is used (<SSEL>=1), select <CSSEL> and <TRGSEL> before the setting of <TBRUN>=<TBPRUN>=1.

Note 2: When the counter is stopped (<TBRUN>="0") and TBxUC<TBUC[15:0]> is read, the value which was captured when the counter was operated is read.

10.4.4 TBxCR(Control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBWBF	R/W	Double buffer 0: Disable 1: Enable
6-5	−	R/W	Write as "0".
4	−	R	Read as "0".
3	I2TB	R/W	Operation at IDLE mode 0: Stop 1: Operation
2	−	R/W	Write as "0".
1	TRGSEL	R/W	External Trigger select 0: Rising edge 1: Falling edge
0	CSSEL	R/W	Counter Start select 0: Software start 1: External trigger

Note 1: Do not modify TBxCR during operating TMRB.

Note 2: When the external trigger start is used (<CSSEL>=1), select <CSSEL> and <TRGSEL> before the setting of <TBRUN>=<TBPRUN>=1.

10.4.5 TBxMOD(Mode register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBRSWR	R/W	Writes to timer registers 0 and 1 (when double buffering is enabled) 0: The data transfer to the timer register 0 and 1 is done by corresponding to the up-counter (UC) regardless of the rewriting of the buffer register 0 and 1. 1: To transfer the buffer registers data to the timer registers, the writing of the timer register 0 and 1 together are needed.
6	TBCP	W	Capture control by software 0: Capture by softwareTB 1: Don't care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as "1".
5-4	TBCPM[1:0]	R/W	Capture timing 00: Disable Capture timing 01: TBxIN↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. 10: TBxIN↑ TBxIN↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN pin input. 11: TIMPLS↑ TIMPLS↓ Takes count values into capture register 0 (TBxCP0) upon rising of TIMPLS input. Takes count values into capture register 1 (TBxCP1) upon falling of TIMPLS input.
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. 00: TBxIN pin input 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

Note: Do not change TBxMOD register while the timer is operating.

10.4.6 TBxFFCR(Flip-flop control register)

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

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

Note: Do not change TBxFFCR register while the timer is operating.

10.4.7 TBxST(Status register)

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

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

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

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

10.4.8 TBxIM(Interrupt mask register)

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

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

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

10.4.9 TBxUC(Up counter capture register)

	31	30	29	28	27	26	25	24
	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	bit symbol 0	0	0	0	0	0	0	0

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

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

10.4.10 TBxRG0(Timer register 0)

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

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

10.4.11 TBxRG1(Timer register 1)

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

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

10.4.12 TBxCP0(Capture register 0)

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

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

10.4.13 TBxCP1(Capture register 1)

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

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

10.5 Description of Operations for Each Circuit

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

10.5.1 Prescaler

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

The prescaler input clock φT0 is 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 10-2 show prescaler output clock resolutions.

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

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.05 μs)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)
		001 (fperiph/2)	fc/2^2 (0.10 μs)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)
		010 (fperiph/4)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		011 (fperiph/8)	fc/2^4 (0.4μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		100 (fperiph/16)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		101 (fperiph/32)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^2 (0.1 μs)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)
		001 (fperiph/2)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		010 (fperiph/4)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		011 (fperiph/8)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		100 (fperiph/16)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		101 (fperiph/32)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^3 (0.2 μs)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)
		001 (fperiph/2)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		010 (fperiph/4)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		011 (fperiph/8)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^4 (0.4 μs)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)
		001 (fperiph/2)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		010 (fperiph/4)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		011 (fperiph/8)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		100 (fperiph/16)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		101 (fperiph/32)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^5 (0.8 μs)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^6 (1.6 μs)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)
		010 (fperiph/4)	fc/2^7 (3.2 μs)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)
		011 (fperiph/8)	fc/2^8 (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
		100 (fperiph/16)	fc/2^9 (12.8 μs)	fc/2^{11} (51.2 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)	fc/2^{14} (409.6 μs)

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

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.05 μs)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.10 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	100 (fc/2)	000 (fperiph/1)	−	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	$fc/2^2$ (0.10 μs)	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	−	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	−	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	$fc/2^3$ (0.2 μs)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	110 (fc/8)	000 (fperiph/1)	−	−	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	−	$fc/2^4$ (0.4 μs)	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	−	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	$fc/2^4$ (0.4μs)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	111 (fc/16)	000 (fperiph/1)	−	−	$fc/2^5$ (0.8 μs)
		001 (fperiph/2)	−	−	$fc/2^6$ (1.6 μs)
		010 (fperiph/4)	−	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)
		011 (fperiph/8)	−	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)
		100 (fperiph/16)	$fc/2^5$ (0.8 μs)	$fc/2^7$ (3.2 μs)	$fc/2^9$ (12.8 μs)
		101 (fperiph/32)	$fc/2^6$ (1.6 μs)	$fc/2^8$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).
Note 2: Do not change the clock gear while the timer is operating.
Note 3: "−" denotes a setting prohibited.

Table 10-3 Prescaler Output Clock Resolutions (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/2^6 (1.6 μs)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		010 (fperiph/4)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		011 (fperiph/8)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		100 (fperiph/16)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		001 (fperiph/2)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		010 (fperiph/4)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		011 (fperiph/8)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		100 (fperiph/16)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
		101 (fperiph/32)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		001 (fperiph/2)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		010 (fperiph/4)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		011 (fperiph/8)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
		100 (fperiph/16)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)
		101 (fperiph/32)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)	fc/2^{16} (1638.4 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		001 (fperiph/2)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		010 (fperiph/4)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
		011 (fperiph/8)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)
		100 (fperiph/16)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)	fc/2^{16} (1638.4 μs)
		101 (fperiph/32)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)	fc/2^{16} (1638.4 μs)	fc/2^{17} (3276.8 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		001 (fperiph/2)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
		010 (fperiph/4)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)
		011 (fperiph/8)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)	fc/2^{16} (1638.4 μs)
		100 (fperiph/16)	fc/2^{14} (409.6μs)	fc/2^{15} (819.2 μs)	fc/2^{16} (1638.4 μs)	fc/2^{17} (3276.8 μs)
		101 (fperiph/32)	fc/2^{15} (819.2 μs)	fc/2^{16} (1638.4 μs)	fc/2^{17} (3276.8 μs)	fc/2^{18} (6553.6 μs)

Table 10-3 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/2^6 (1.6 μs)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		010 (fperiph/4)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		011 (fperiph/8)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		100 (fperiph/16)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^6 (1.6 μs)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		010 (fperiph/4)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		011 (fperiph/8)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		100 (fperiph/16)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^6 (1.6 μs)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		010 (fperiph/4)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		011 (fperiph/8)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		100 (fperiph/16)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^6 (1.6 μs)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		010 (fperiph/4)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		011 (fperiph/8)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		100 (fperiph/16)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^6 (1.6 μs)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)
		001 (fperiph/2)	fc/2^7 (3.2 μs)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)
		010 (fperiph/4)	fc/2^8 (6.4 μs)	fc/2^9 (12.8 μs)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)
		011 (fperiph/8)	fc/2^9 (12.8μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4μs)
		100 (fperiph/16)	fc/2^{10} (25.6μs)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)
		101 (fperiph/32)	fc/2^{11} (51.2μs)	fc/2^{12} (102.4 μs)	fc/2^{13} (204.8 μs)	fc/2^{14} (409.6μs)

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.

10.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

  UC source clock, specified by TBxMOD<TBCLK[2:0]>, can be selected from either three types φT1, φT4, φT16 ,φT32, φT64, φT128,φT256of prescaler output clock or the external clock of the TBxIN 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 INTTBx0 overflow interrupt is generated.

10.5.3 Timer registers (TBxRG0, TBxRG1)

TBxRG0 and TBxRG1 are registers for setting values to compare with up-counter values and two registers are built into each channel. If the comparator detects a match between a value set in this timer register and that in a UC up-counter, it outputs the match detection signal.

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

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

10.5.4 Capture

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

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

10.5.5 Capture register (TBxCP0, TBxCP1)

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

10.5.6 Up counter capture register (TBxUC)

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

10.5.7 Comparators (CP0, CP1)

This register compares with the up-counter (UC) and the value setting of the Timer Register (TBxRG0 and TBxRG1) to detect whether there is a match or not. If a match is detected, INTTBx0 and INTTBx1 are generated.

10.5.8 Timer Flip-flop (TBxFF0)

The timer flip-flop (TBxFF0) is reversed by a match signal from the comparator and a latch signal to the capture registers. It can be enabled or disabled to reverse by setting the TBxFFCR<TBC1T1, TBC0T1, TBE1T1, TBE0T1>.

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

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

10.5.9 Capture interrupt (INTCAPx0, INTCAPx1)

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

10.6 Description of Operations for Each Mode

10.6.1 16-bit Interval Timer Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx1 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, 010, 011,100,101,110,111)								
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don't care −; No change

10.6.2 16-bit Event Counter Mode

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

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Set PORT registers.										Allocates corresponding port to TBxIN.
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger.
TBxMOD	←	0	1	0	0	0	0	0	0	Changes to TBxIN as an input clock.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.
TBxMOD	←	0	0	0	0	0	0	0	0	Software capture is done.

Note: X; Don't care −; No change

10.6.3 16-bit PPG (Programmable Pulse Generation) Output Mode

Square waves with any frequency and any duty (programmable square waves) can be output. The output pulse can be either low-active or high-active.

Programmable square waves can be output from the TBxOUT pin by triggering the timer flip-flop (TBxFF0) to reverse when the set value of the up-counter (UC) matches the set values of the timer registers (TBxRG0 and TBxRG1). Note that the set values of TBxRG0 and TBxRG1 must satisfy the following requirement:

Set value of TBxRG0 < Set value of TBxRG1

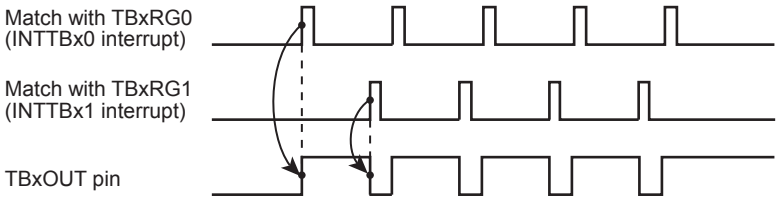


Figure 10-2 Example of Output of Programmable Pulse Generation (PPG)

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

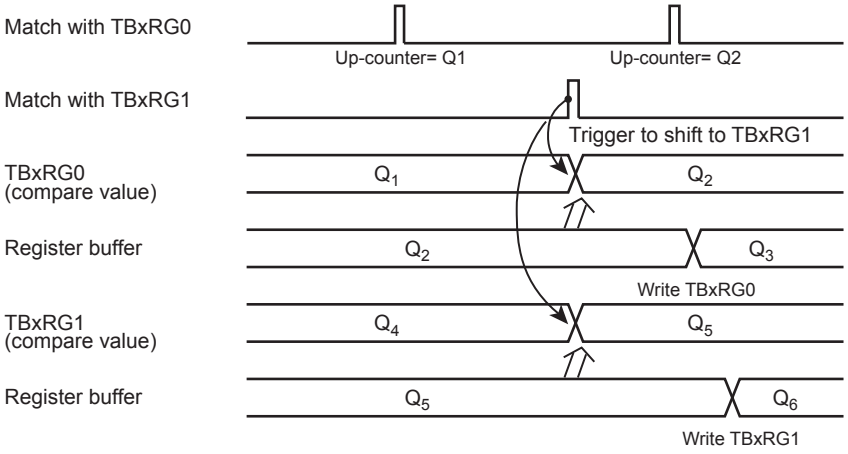


Figure 10-3 Register Buffer Operation

The block diagram of this mode is shown below.

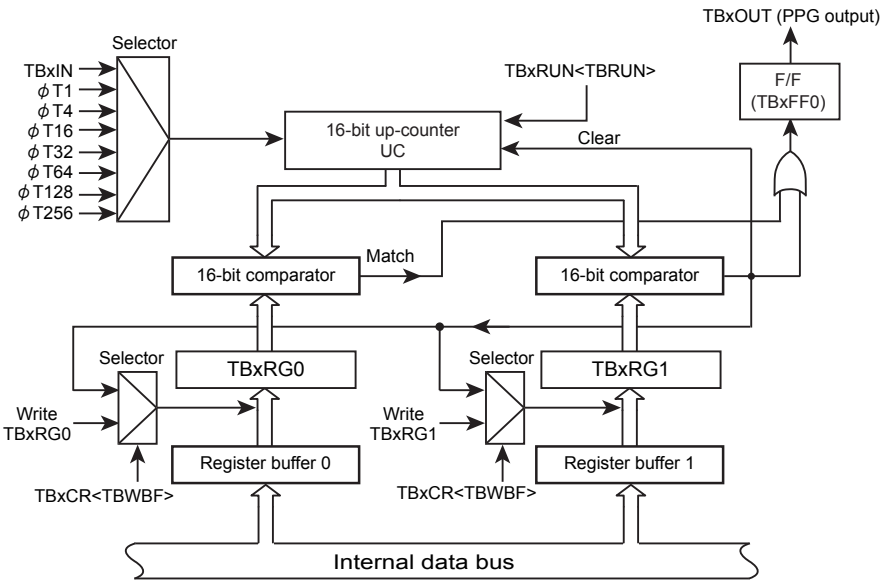


Figure 10-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	0	X	−	0	0	0	Disables double buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	0	X	−	X	0	0	Enables double buffering.
										(Changes the duty/cycle when the INTTBx0 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, 010, 011,100,101,110,111)					
										UC is cleared to match TBxRG1.
Set PORT registers.										Allocates corresponding port to TBxOUT.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don't care
−; No change

10.6.4 External trigger Programmable Pulse Generation Output Mode (PPG)

Using an external count start trigger enables one-shot pulse generation with a short delay.

The 16-bit up-counter (UC) is programmed to count up on the rising edge of the TBxIN pin (TBxCR[1:0] = "01"). The TBxRG0 is loaded with the pulse delay (d), and the TBxRG1 is loaded with the sum of the TBxRG0 value (d) and the pulse width (p). The above settings must be done while the 16-bit up-counter is stopped (TBxRUN<TBRUN> = 0).

To enable the trigger for timer flip-flop, sets TBxFFCR<TBE1T1, TBE0T1> to "11". With this setting, the timer flip-flop reverses when 16-bit up-counter (UC) corresponds to TBxRG0 or TBxRG1.

Sets TBxRUN<TBRUN> to "1" to enable the count-up by an external trigger.

After the generation of one-shot pulse by the external trigger, to disable reverse of the timer flip-flop or to stop 16bit counter by TBxRUN<TBRUN> setting.

Symbols (d) and (p) used in the text correspond to symbols d and p in Figure 10-5.

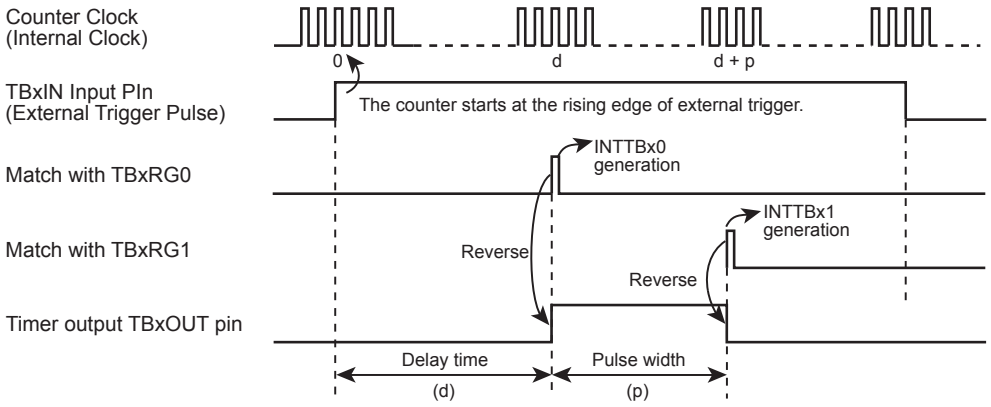


Figure 10-5 One-shot pulse generation using an external count start trigger (with a delay)

10.7 Applications using the Capture Function

The capture function can be used to develop many applications, including those described below:

1. One-shot pulse output triggered by an external pulse
2. Pulse width measurement

10.7.1 One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TBxIN 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 INTTBx0 / INTTBx1 interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in Figure 10-6.

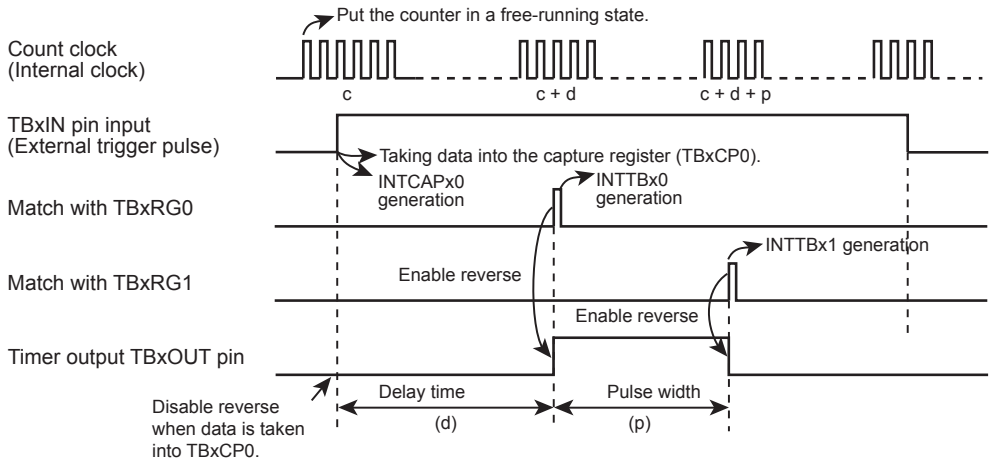


Figure 10-6 One-shot Pulse Output (With Delay)

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

		7	6	5	4	3	2	1	0	
[[Main processing] Capture setting by TBxIN										
Set PORT registers.										Allocates corresponding port to TBxIN.
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxMOD	←	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 TBxIN.
TBxFFCR	←	X	X	0	0	0	0	1	0	Clears TBxFF0 reverse trigger and disables.
Set PORT registers.										Allocates corresponding port to TBxOUT.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupts specified by INTCAPx0 interrupt corresponding bit by setting to "1".
TBxRUN	←	*	*	*	*	*	1	X	1	Starts the TMRBx module.
[Processing of INTCAPx0 interrupt service routine] Pulse output setting										
TBxRG0	←	*	*	*	*	*	*	*	*	Sets count value.(TBxCP0 + 3ms/ΦT1)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Sets count value.(TBxCP0 + (3+2)ms/ΦT1)
	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	−	−	1	1	−	−	Reverses TBxFF0 if UC consistent with TBxRG0 and TBxRG1.
TBxIM	←	X	X	X	X	X	1	0	1	Masks except TBxRG1 correspondence interrupt.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupt specified by INTTBx1 interrupt corresponding bit setting to "1".
[Processing of INTTBx1 interrupt service routine] Output disable										
TBxFFCR	←	X	X	−	−	0	0	−	−	Clears TBxFF0 reverse trigger setting.
	←	*	*	*	*	*	*	*	*	Prohibits interrupts specified by INTTBx interrupt corresponding bit by setting to "1".

Note: 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 INTTBx1 interrupt.

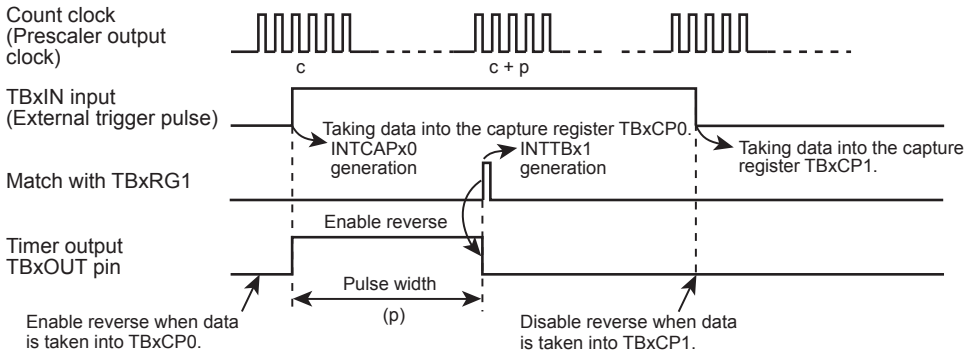


Figure 10-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

10.7.2 Pulse width measurement

By using the capture function, the "High" level width of an external pulse can be measured. Specifically, by putting it in a free-running state using the prescaler output clock, an external pulse is input through the TBxIN pin and the up-counter (UC) is made to count up. A trigger is generated at each rising and falling edge of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TBxCP0, TBxCP1). The CPU must be programmed so that INTCAPx1 is generated at the falling edge of an external pulse input through the TBxIN pin.

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

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

Caution must be exercised when measuring pulse 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 10-8 and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

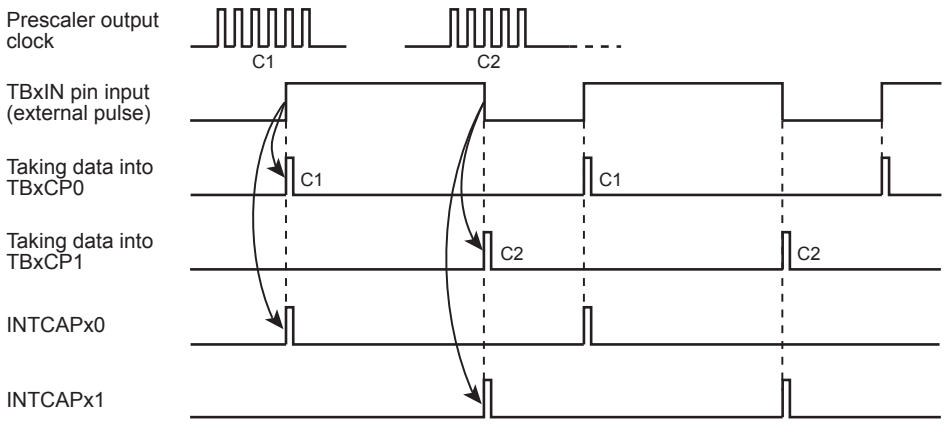


Figure 10-8 Pulse Width Measurement

11. Serial Channel (SIO/UART)

11.1 Overview

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

- Synchronous communication mode (I/O interface mode)
- Asynchronous communication mode (UART mode)

Their features are given in the following.

- Transfer Clock
  - Dividing by the prescaler, from the peripheral clock (φT0) frequency into 1/1, 1/2, 1/4,1/8, 1/16, 1/32, 1/64,1/128.
  - Make it possible to divide from the prescaler output clock frequency into 1 to 16.
  - Make it possible to divide from the prescaler output clock frequency into N+m/16 (N=2 to 15, m=1 to 15). (only UART mode)
  - The usable system clock (only UART mode).
- Buffer
  - The usable double buffer function.
  - Make it possible to clear the transmit buffer.
- FIFO

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

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

    Keep a "High" level/ "Low" level
  - The last bit hold time of TXDx pin can be set in the case that SCLK is input.
- UART Mode
  - Data length: 7 bits, 8bits, 9bits
  - Add parity bit (to be against 9bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{CTSx}}$ pin

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

Note: In this chapter, if there is no description of the channel number part "x" such as terminal name, register name etc., please read it with "x" as necessary.

11.2 Difference in the Specifications of SIO Modules

TMPM3U0FSDMG has two SIO channels.

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

Table 11-1 Difference in the Specifications of SIO Modules

	Pin name			Interrupt		UART source clock
	TXD	RXD	$\overline{\text{CTS}}$x/ SCLKx	Receive Interrupt	Transmit Interrupt	
Channel 0	PE0	PE1	PE2	INTRX0	INTTX0	TB4OUT
Channel 1	PF0	PB6	-	INTRX1	INTTX1	TB4OUT
	PB4		-			

11.3 Configuration

Figure 11-1 shows Serial channel block diagram.

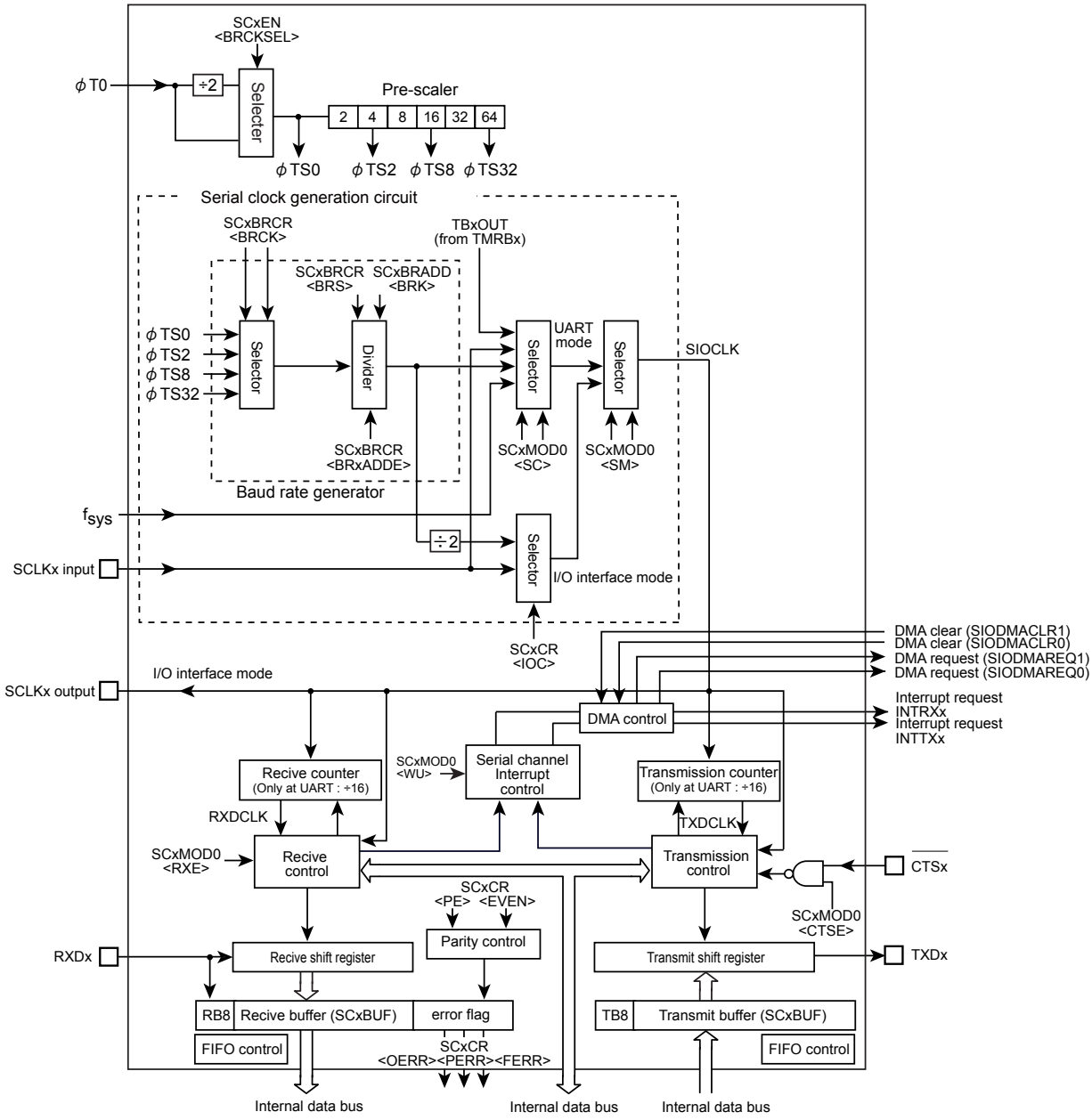


Figure 11-1 Serial Channel Block Diagram

11.4 Registers Description

11.4.1 Registers List

The table below shows control registers and their addresses.

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

Channel x	Base Address
Channel0	0x4002_0080
Channel1	0x4002_00C0

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

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

11.4.2 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	-	-	-	-	-	-	BRCKSEL	SIOE
After reset	0	0	0	0	0	0	0	0

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

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

11.4.4 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	-	EHOLD			-	TXDEMP	TIDLE	
After reset	0	0	0	0	0	1	1	0
	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
After reset	0	0	0	0	0	0	0	0

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

Bit	Bit Symbol	Type	Function
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selecting input clock edge (For I/O Interface mode) Set to "0" in the clock output mode. 0:Data in the transmit buffer is sent to TXDx pin one bit at a time on the falling edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the rising edge of SCLKx. In this case, the SCLKx starts from high level. 1:Data in the transmit buffer is sent to TXDx pin one bit at a time on the rising edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the falling edge of SCLKx. In this case, the SCLKx starts from low level.
0	IOC	R/W	Selecting clock (For I/O Interface mode) 0: Baud rate generator 1: SCLKx pin input

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

11.4.5 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-bit UART mode.
6	CTSE	R/W	Handshake function control (For UART) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{CTS}$ pin.
5	RXE	R/W	Receive control (Note1)(Note2) 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 output 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLKx pin input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

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

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

11.4.6 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. And when FIFO is enabled, specify the configuration of FIFO. In UART mode, specify the only configuration of FIFO.
4	TXE	R/W	Transmit control (Note1)(Note2) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface mode) 000: None 001: 1 x SCLK cycle 010: 2 x SCLK cycle 011: 4 x SCLK cycle 100: 8 x SCLK cycle 101: 16 x SCLK cycle 110: 32 x SCLK cycle 111: 64 x SCLK cycle This parameter is valid only for the I/O interface mode when SCLK pin output is selected. In other modes, this parameter has no meaning. Specifies the interval time of continuous transmission when double buffering is enabled in the I/O interface mode.
0	−	R/W	Write a "0".

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

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

11.4.7 SCxMOD2 (Mode Control Register 2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBEMP	R	Transmit buffer empty flag. 0: Full 1: Empty If double buffering is disabled, this flag is insignificant. This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1". Writing data again to the double buffers sets this bit to "0".
6	RBFLL	R	Receive buffer full flag. 0: Empty 1: Full If double buffering is disabled, this flag is insignificant. This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0".
5	TXRUN	R	In transmission flag 0: Stop 1: Operate This is a status flag to show that data transmission is in progress. <TXRUN> and <TBEMP> bits indicate the following status.
4	SBLEN	R/W	STOP bit length (for UART mode) 0 : 1-bit 1 : 2-bit This specifies the length of transmission stop bit in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.
3	DRCHG	R/W	Setting transfer direction 0: LSB first 1: MSB first Specifies the direction of data transfer in the I/O interface mode. In the UART mode, set this bit to LSB first.
2	WBUF	R/W	Enable double-buffer 0: Disabled 1 : Enabled This parameter enables or disables the transmit/receive double buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data I/O interface mode and to transmit data in the UART mode. When receiving data in the I/O interface mode (SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits are initialized and the transmit/receive circuit become initial state (Note1)(Note2).
<table>
<tr><th>Register</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td>RXE</td></tr>
<tr><td>SCxMOD1</td><td>TXE</td></tr>
<tr><td>SCxMOD2</td><td>TBEMP, RBFLL, TXRUN</td></tr>
<tr><td>SCxCR</td><td>OERR, PERR, FERR</td></tr>
</table></td></tr>
</table>

Note 1: While data transmission is in progress, any software reset operation must be executed twice in succession.

Note 2: A software reset requires 2 clocks-duration at the time between the end of recognition and the start of execution of software reset instruction.

11.4.8 SCxBRCR (Baud Rate Generator Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Write "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: φTS0 01: φTS2 10: φTS8 11: φTS32
3-0	BRS[3:0]	R/W	Division ratio "N" 0000: 16 0001: 1 0010: 2 ... 1111: 15

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

Note 2: The division ratio "1" of the baud rate generator can be specified only when the double buffering is used in the I/O interface mode.

11.4.9 SCxBRADD (Baud Rate Generator Control Register 2)

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

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

Table 11-2 lists the settings of baud rate generator division ratio.

Table 11-2 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.

11.4.10 SCxFCNF (FIFO Configuration Register)

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

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

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

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

11.4.11 SCxRFC (Receive FIFO Configuration Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RFCS	W	Receive FIFO clear (Note1) 1: Clear When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL[2:0]> is "000". And also the read pointer is initialized. Read as "0".
6	RFIS	R/W	Select interrupt generation condition. 0: When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt <RIL[1:0]> 1: When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt <RIL[1:0]>
5-2	−	R	Read as "0".
1-0	RIL[1:0]	R/W	FIFO fill level to generate Receive interrupts.

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

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

11.4.12 SCxTFC (Transmit FIFO Configuration Register)

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

Bit	Bit Symbol	Type	Function
31-9	−	R	Read as "0".
8	TBCLR	W	Transmit buffer clear 1: Clear When SCxTST<TBCLR> is set to "1", the transmit buffer is cleared. Read as "0".
7	TFCS	W	Transmit FIFO clear (Note1) 1: Clear When SCxTST<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL[2:0]> is "000". And also the write pointer is initialized. Read as "0".
6	TFIS	R/W	Selects interrupt generation condition. 0: When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt <TIL [1:0]> 1: When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt <TIL [1:0]>
5-2	−	R	Read as "0".
1-0	TIL[1:0]	R/W	Fill level which transmit interrupt is occurred.

	Half duplex	Full duplex
00	Empty	Empty
01	1 byte	1 byte
10	2 bytes	Empty
11	3 bytes	1 byte

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

Note 2: **In case that SCxEN<SIOE>="0" (Stop SIO/UART operation) or the operation mode is changed to IDLE mode with SCxMOD1<I2SC>="0" (Stop SIO/UART operation in IDLE mode), SCxTFC is initialized again. After you perform the following operations, configure the SCxTFC register again.**

11.4.13 SCxRST (Receive FIFO Status Register)

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

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

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

11.4.14 SCxTST (Transmit FIFO Status Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TUR	R	Transmit FIFO Under run. (Note1) 0: Not generated 1: Generated
6-3	−	R	Read as "0".
2-0	TLVL[2:0]	R	Status of Transmit FIFO level 000: Empty 001: 1 byte 010: 2 byte 011: 3 byte 100: 4 byte

Note 1: <TUR> is cleared to "0" when transmit data is written to the SCxBUF.

11.5 Operation in Each Mode

Table 11-3 shows the modes and data formats.

Table 11-3 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 (I/O 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 SCLKx. SCLKx 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.

11.6 Data Format

11.6.1 Data Format List

Figure 11-2 shows data format.

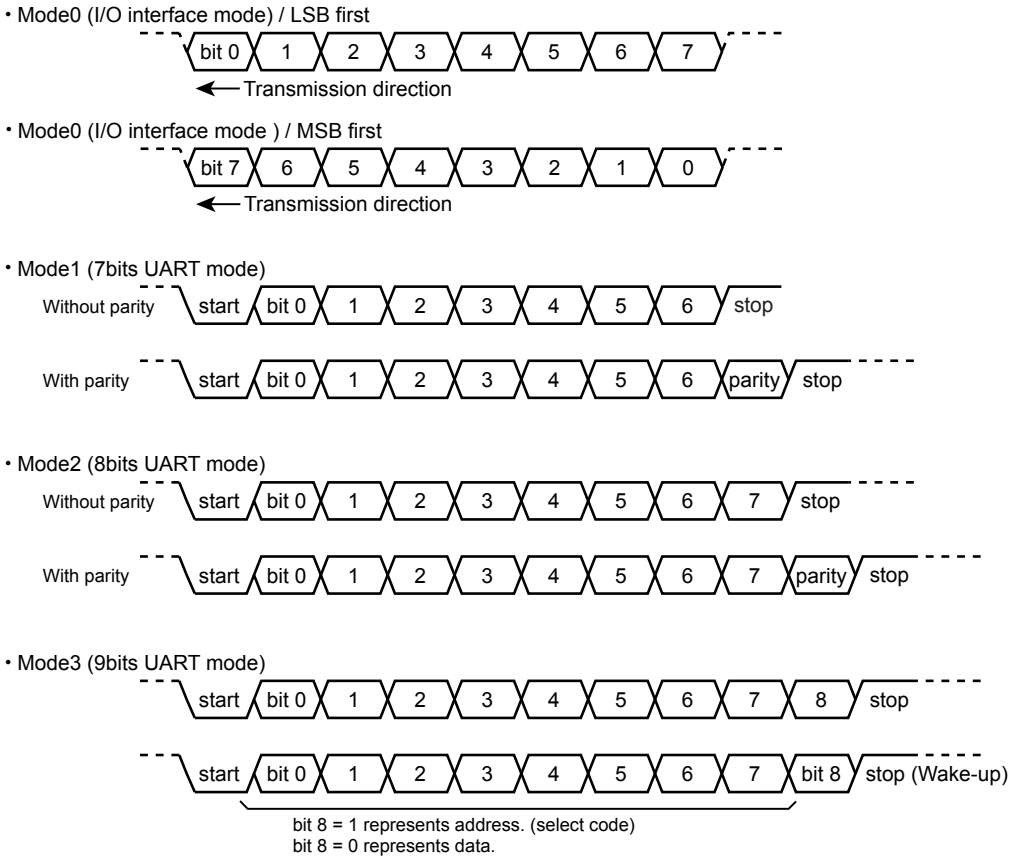


Figure 11-2 Data Format

11.6.2 Parity Control

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

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

SCxCR<EVEN> selects either even or odd parity.

11.6.2.1 Transmission

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

The parity bit will be stored in SCxBUF<TB[7]> 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.

11.6.2.2 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<RB[7]>, in the 8-bit UART mode, it is compared with the one in SCxCR<RB8>.

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

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

11.6.3 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.

11.7 Clock Control

The following figure shows the serial clock (SIOCLK) generation circuit. Before changing the serial clock setting, check if the setting satisfies AC electrical characteristics.

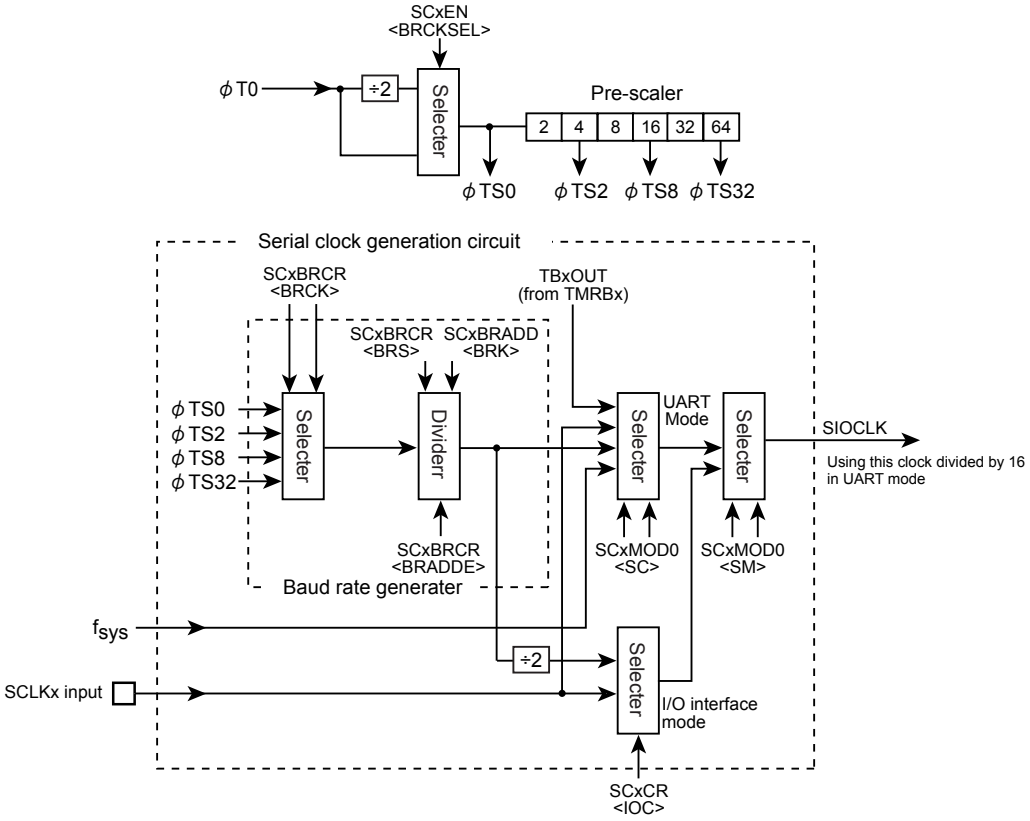


Figure 11-3 Serial clock generation circuit

11.7.1 Prescaler

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

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

11.7.2 Serial Clock Generation Circuit

The serial clock generation circuit is a block to generate transmit and receive clocks (SIOCLK) and consists of the circuits in which clocks can be selected by the settings of the baud rates generator and modes.

11.7.2.1 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 1, 4, 16 and 64.

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

SCxEN<BRCKSEL>		SCxBRCR<BRCK>		Baud rate generator input clock φTx
0	φT0/2	00	φTS0	φT0/2
0		01	φTS2	φT0/8
0		10	φTS8	φT0/32
0		11	φTS32	φT0/128
1	φT0	00	φTS0	φT0
1		01	φTS2	φT0/4
1		10	φTS8	φT0/16
1		11	φTS32	φT0/64

(2) Baud Rate Generator output clock

The frequency division ratio of the output clock in the baud rate generator is set by SCxBRCR and SCxBRADD.

The following frequency divide ratios can be used; 1/N frequency division in the I/O interface mode, either 1/N or 1/(N + (16-K)/16) in the UART mode.

The table below shows the frequency division ratio which can be selected.

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BRS>	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.

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

- Divide by N

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

- N + (16-K)/16 division

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

11.7.2.2 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 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 11-4 shows clock selection in I/O interface mode.

Table 11-4 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	SCLKx pin input rising edge
		Falling edge	SCLKx pin input falling edge

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

- If double buffer is used
  - SCLK cycle > 6/fsys
- If double buffer is not used
  - SCLK cycle > 8/fsys

(2) Transfer clock in the UART mode

Table 11-5 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 11-5 Clock Selection in UART Mode

Mode SCxMOD0<SM>	Clock selection SCxMOD0<SC>
UART Mode	Timer output
	Baud rate generator
	fsys
	SCLKx input

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

- SCLK cycle > 2/fsys

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

Baud rates can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[1:0]>}}{(\text{TBxRG1} \times 2) \times 2 \times 16}$$

In the case the timer prescaler clock ΦT1 (2division ratio) is selected.

One clock cycle is a period that the timer flip-flop is inverted twice.

11.7.3 Transmit/Receive Buffer and FIFO

11.7.3.1 Configuration

Figure 11-4 shows the configuration of transmit buffer, receive buffer and FIFO.

Appropriate settings are required for using buffer and FIFO. The configuration may be predefined depending on the mode.

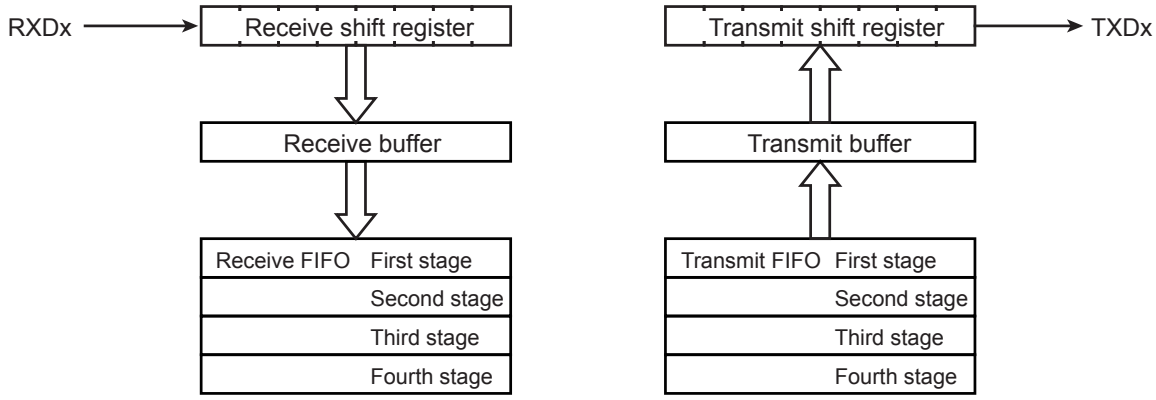


Figure 11-4 The Configuration of Buffer and FIFO

11.7.3.2 Transmit/Receive Buffer

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

In the case of a receive mode, if SCLKx input is set 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 11-6 shows correlation between modes and buffers.

Table 11-6 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

11.7.3.3 Initialize Transmit Buffer

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

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

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

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

11.7.3.4 FIFO

In addition to the double buffer function above described, 4-byte FIFO can be used.

To enable FIFO, enable the double buffer by setting SCxMOD2<WBUF> to "1" and SCxFCNF<CNFG> to "1". The FIFO buffer configuration is specified by SCxMOD1<FDPX[1:0]>.

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

Table 11-7 shows correction between modes and FIFO.

Table 11-7 Mode and FIFO Composition

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

11.8 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".

11.9 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	Over-run error	Parity error	Framing error
I/O Interface (SCLKx input)	Over-run error	Under-run error (When double buffer and FIFO are used)	Fixed to 0
		Fixed to 0 (When a double buffer and FIFO are unused)	
I/O Interface (SCLKx output)	Undefined	Undefined	Fixed to 0

11.9.1 OERR Flag

In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied).

In the I/O interface 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.**

11.9.2 PERR Flag

This flag indicates a parity error in the UART mode and an under-run error or completion of transmit in the I/O interface mode.

In the UART mode, <PERR> is set to "1" when the parity generated from the received data is different from the 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 SCLKx 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 SCLKx 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.

11.9.3 FERR Flag

A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the stop bit length settings in the SCxMOD2<SBLEN>, the stop bit status is determined by only 1.

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

11.10 Receive

11.10.1 Receive Counter

The receive counter is a 4-bit binary counter and is up-counted by SIOCLK.

In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the eighth pulses. A recevived signal issampled by SIOCLK, and after carrying out filtering processing which becomes effective by continuation coincidence 3 times, it is treated as received data.

11.10.2 Receive Control Unit

11.10.2.1 I/O interface mode

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

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

11.10.2.2 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.

11.10.3 Receive Operation

11.10.3.1 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. When the double buffer is disabled, the receive buffer full flag has no meaning.

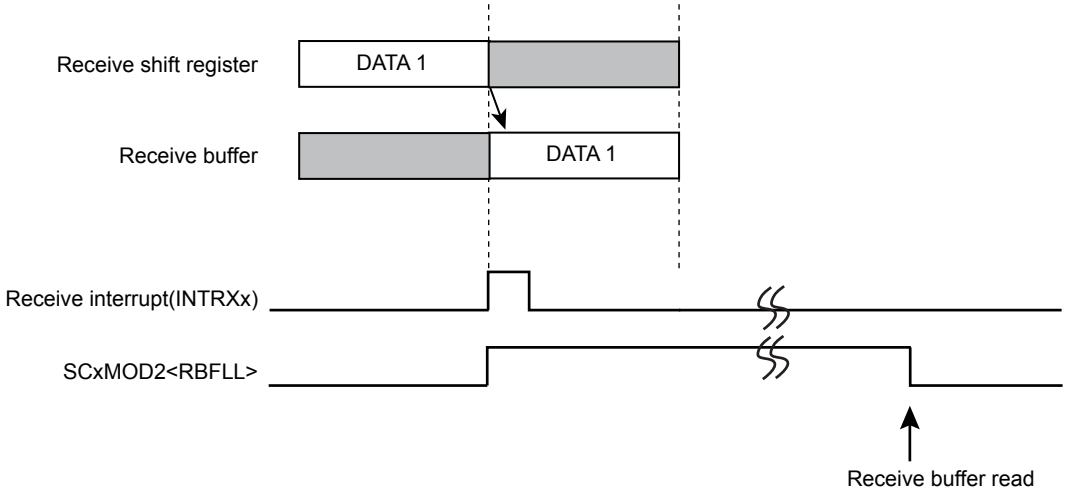


Figure 11-5 Receive Buffer Operation

11.10.3.2 Receive FIFO Operation

When FIFO is enabled, the received data is moved from receive buffer to receive FIFO and the receive buffer full flag is cleared immediately. An interrupt will be generated according to the SCxRFC<RIL[1:0]>.

Note: **When the data with parity bit are received in UART mode by using the FIFO, the parity error flag is shown the occurring the parity error in the received data.**

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

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

After setting of the above FIFO configuration, the data reception is started by writing "1" to the SCxMOD0<RXE>. When the data is stored all in the receive shift register, receive buffer and receive FIFO, SCxMOD0<RXE> is automatically cleared and the receive operations finished.

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

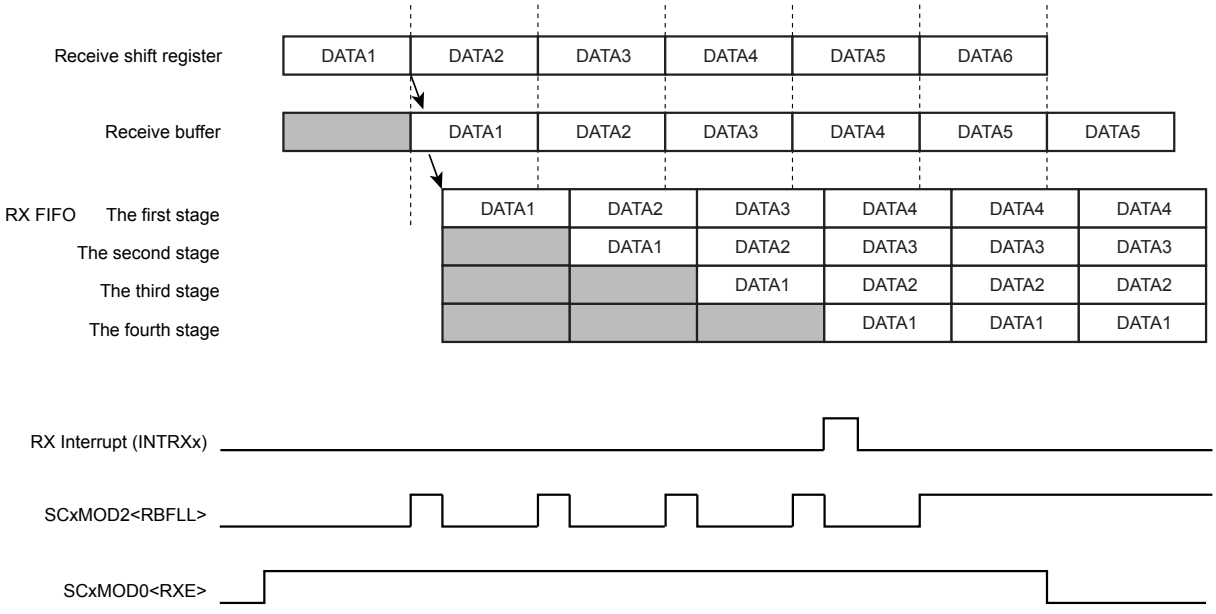


Figure 11-6 Receive FIFO Operation

11.10.3.3 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 over-run error flag has no meaning.

The timing of SCLK output stop and re-output depends on receive buffer and FIFO.

(1) Case of single buffer

Stop 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 a 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 the received buffer and SCLK output restarts.

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

11.10.3.4 Read Received Data

In spite of enabling or disabling FIFO, read the received data from the receive buffer (SCxBUF).

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. The next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

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

11.10.3.5 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".

11.10.3.6 Overrun Error

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

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

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

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

11.11 Transmission

11.11.1 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.

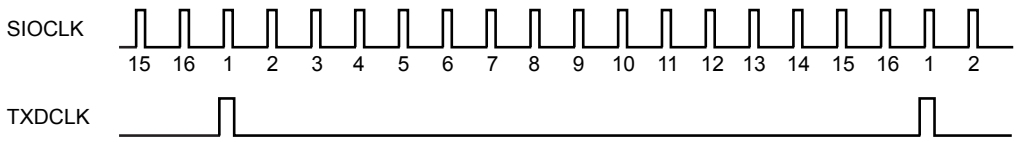


Figure 11-7 Generation of Transmission Clock

11.11.2 Transmission Control

11.11.2.1 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 TXDx pin on the falling edge of SCLKx pin.

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

11.11.2.2 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.

11.11.3 Transmit Operation

11.11.3.1 Operation of Transmission Buffer

If double buffering is disabled, the CPU writes data only to transmit shift register and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled (including the case the transmit FIFO is enabled), data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

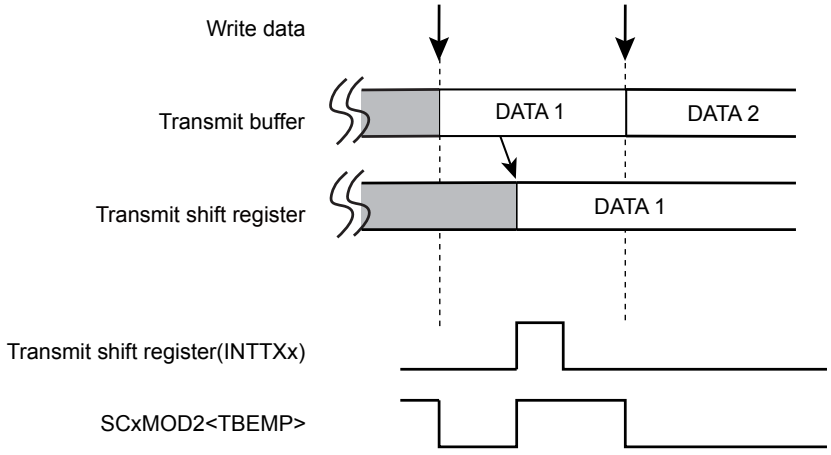


Figure 11-8 Operation of Transmission Buffer (Double-buffer is enabled)

11.11.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

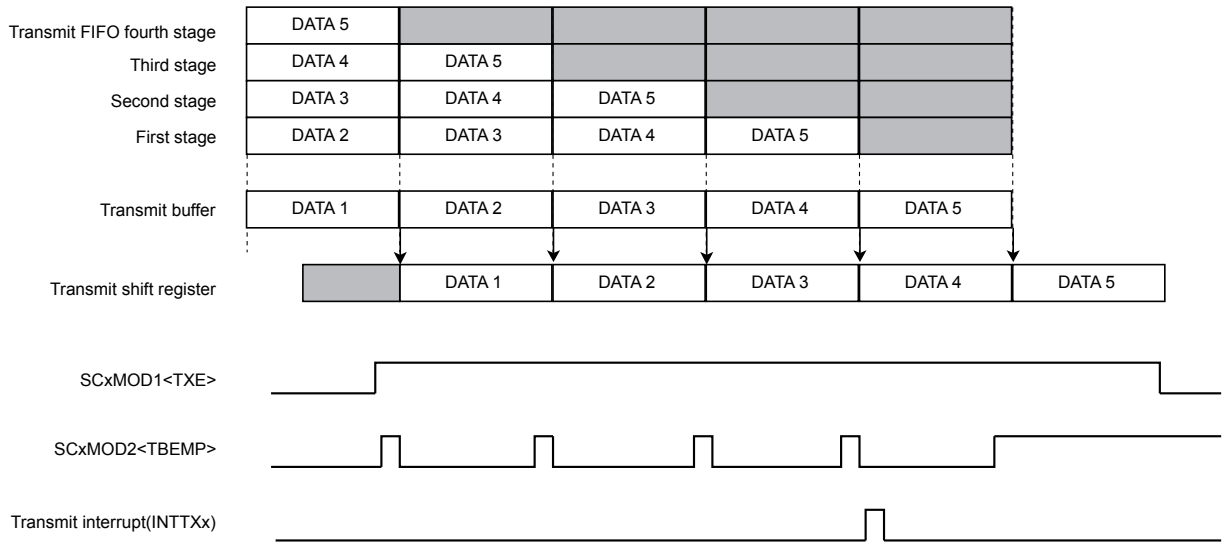
Note: To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").

Settings and operations to transmit 4-byte data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1<FDPX[1:0]> = "10"	:Transfer mode is set to half duplex.
SCxFCNF<RFST><TFIE><RFIE>	:Transmission is automatically disabled if FIFO becomes empty.
<RXTXCNT><CNFG> = "11011"	:The number of bytes to be used in the receive FIFO is the same as the interrupt :generation fill level.
SCxTFC<TIL[1:0]> = "00"	:Sets the interrupt generation fill level to "0".
SCxTFC<TFCS[1:0]> = "11"	:Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF<CNFG> = "1"	:Enable FIFO

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

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts writing transmit data.



11.11.3.3 I/O interface Mode/Transmission by SCLK Output

In the I/O interface mode and SCLK output setting, 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 in 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.

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

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

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

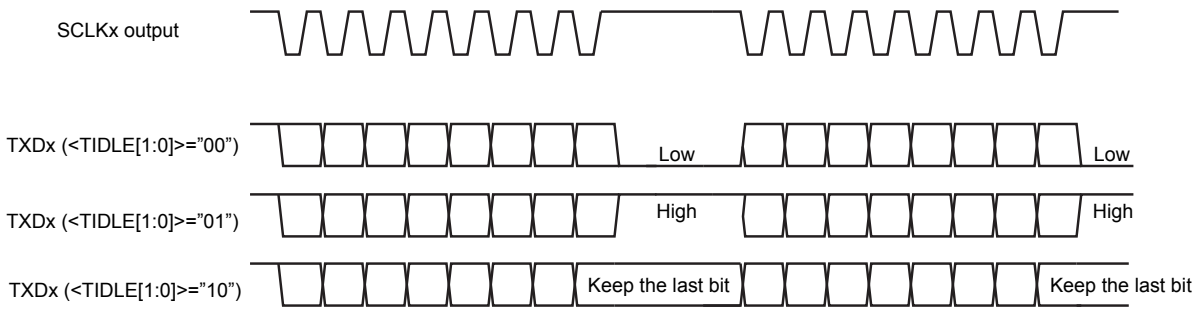


Figure 11-9 Level of TXDx pin After the last bit is output

11.11.3.5 Under-run error

In the I/O interface SCLK input mode and if FIFO has no data and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

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

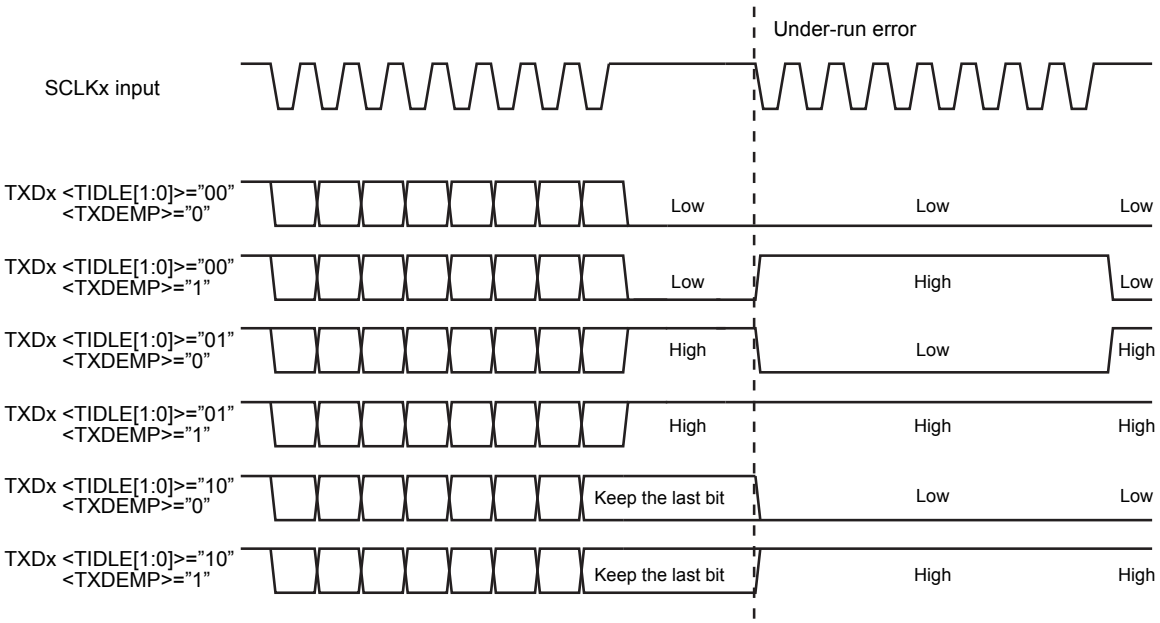


Figure 11-10 Level of TXDx pin when Under-run Error is Occurred

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

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

11.11.3.6 Data Hold Time In the I/O interface SCLK input mode

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

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

11.12 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 over-run errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTSx}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTSx}}$ pin returns to the "Low" level. The INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note 1: If the $\overline{\text{CTS}}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.

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

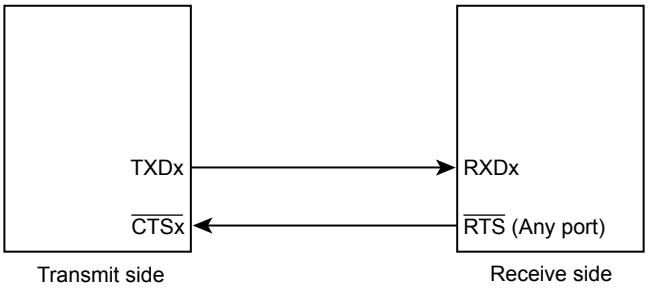


Figure 11-11 Handshake Function

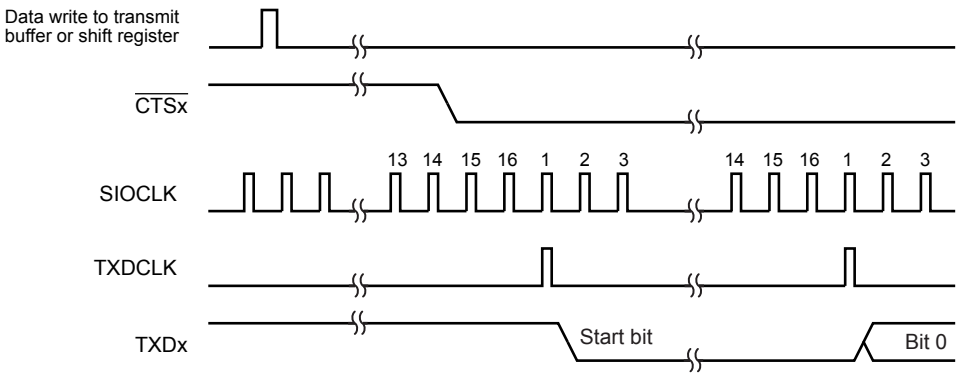


Figure 11-12 $\overline{\text{CTSx}}$ Signal timing

11.13 Interrupt/Error Generation Timing

11.13.1 Receive Interrupts

Figure 11-13 shows the data flow of receive operation and the route of read.

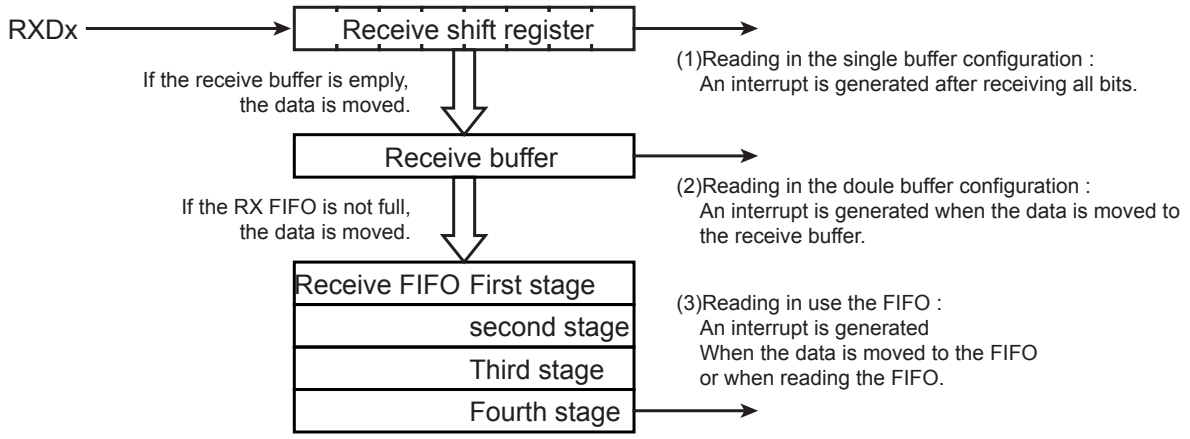


Figure 11-13 Receive Buffer/FIFO Configuration Diagram

11.13.1.1 Single Buffer / Double Buffer

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

Buffer Configurations	UART modes	IO interface modes
Single Buffer	−	Immediately after the raising / falling edge of the last SCLKx pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	Around the center of the first stop bit	Immediately after the raising / falling edge of the last SCLKx pin (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 over-run error is occurred.

11.13.1.2 FIFO

In use of FIFO, receive interrupt is generated on the condition that the following either operation and SCxRFC<RFIS> setting are established.

- When transfer a received data from receive buffer to receive FIFO
- When read a receive data from receive FIFO

Interrupt conditions are decided by the SCxRFC<RFIS> settings as described in Table 11-8.

Table 11-8 Receive Interrupt Conditions in use of FIFO

SCxRFC<RFIS>	Interrupt conditions
"0"	When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt <RIL[1:0]>
"1"	When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt <RIL[1:0]>

11.13.2 Transmit interrupts

Figure 11-14 shows the data flow of transmit operation and the route of read.

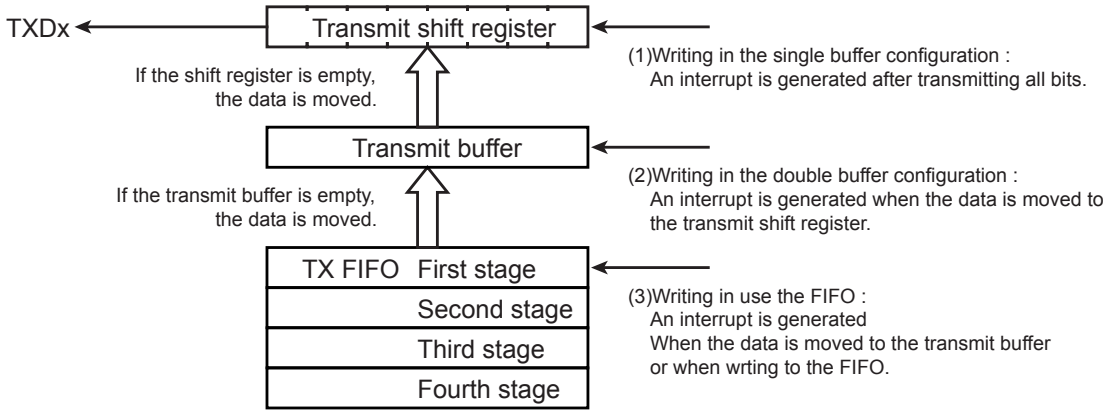


Figure 11-14 Transmit Buffer / FIFO Configuration Diagram

11.13.2.1 Singe Buffer / Double Buffer

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

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCLKx pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register. In case of transmit shift register is empty, transmit interrupt is generated not depend on SCxMOD1<TXE> because a data written to transfer buffer is moved from transmit buffer to transmit shift register.	

11.13.2.2 FIFO

In use of FIFO, transmit interrupt is generated on the condition that the following either operation and SCxTFC<TFIS> setting are established.

- When transmitted data is transferred from transmit FIFO to transmit buffer
- When transmit data is write into transmit FIFO

Interrupt conditions are decided by the SCxTFC<TFIS> settings as described in Table 11-9.

Table 11-9 Transmit Interrupt conditions in use of FIFO

SCxTFC<TFIS>	Interrupt condition
"0"	When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>
"1"	When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>

11.13.3 Error Generation

11.13.3.1 UART Mode

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

11.13.3.2 I/O Interface Mode

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

11.14 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01".

As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR

<OERR><PERR><FERR> are initialized. And the receive circuit and the transmit circuit become initial state. Other states are maintained.

11.15 Operation in Each Mode

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

11.15.1.1 Transmitting Data

(1) SCLK Output Mode

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

  Data is output from the TXDx pin and the clock is output from the SCLKx pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.
- If the transmit double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer while data transmission is halted or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  When data is moved from the transmit buffer to the transmit shift register, if the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the SCLK output stops.

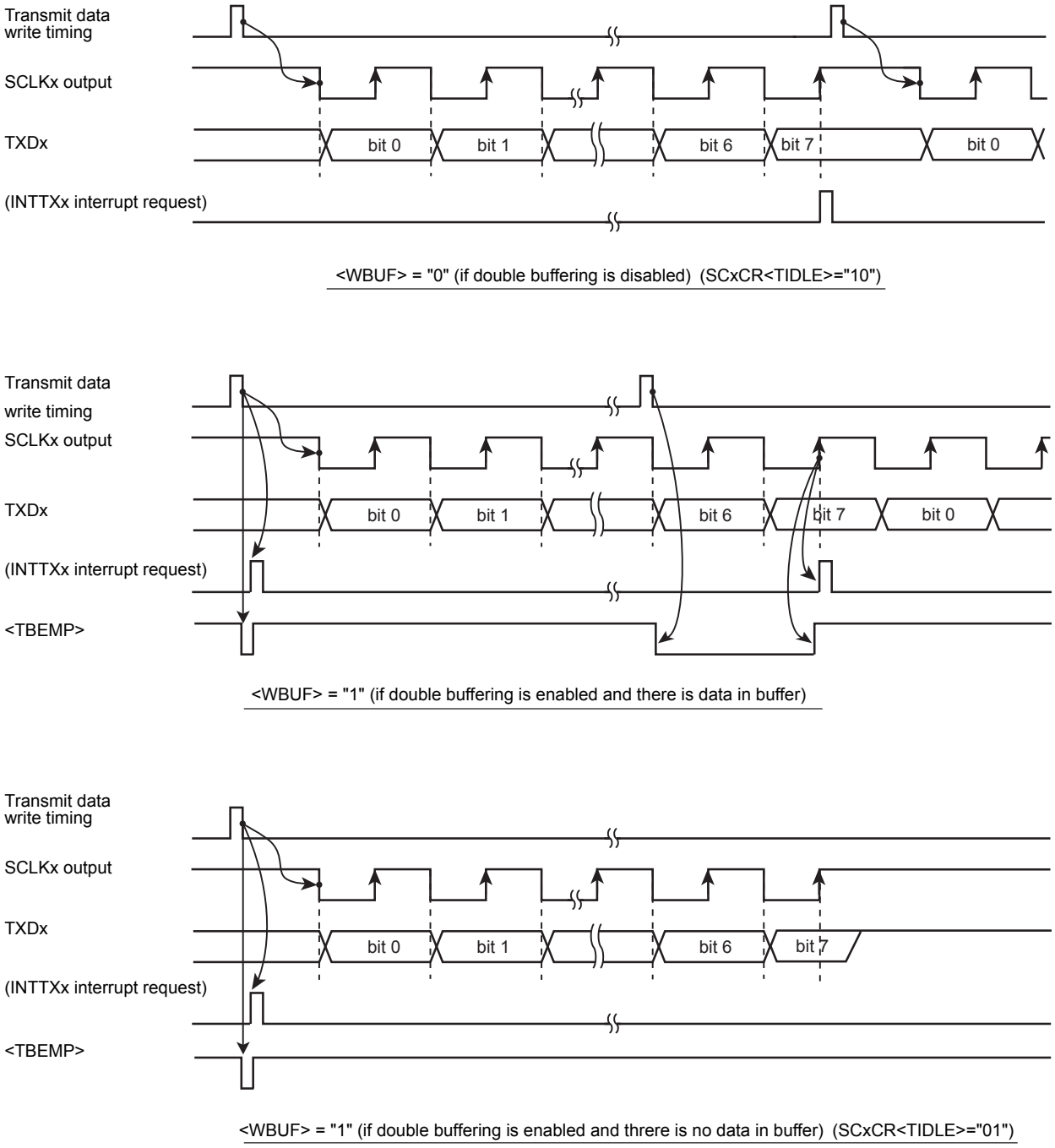


Figure 11-15 Transmit Operation in the I/O Interface Mode (SCLK Output Mode)

(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 TXDx 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 11-16.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the 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 the level which is specified by SCxCR<TXDEMP> is output to TXDx pin.

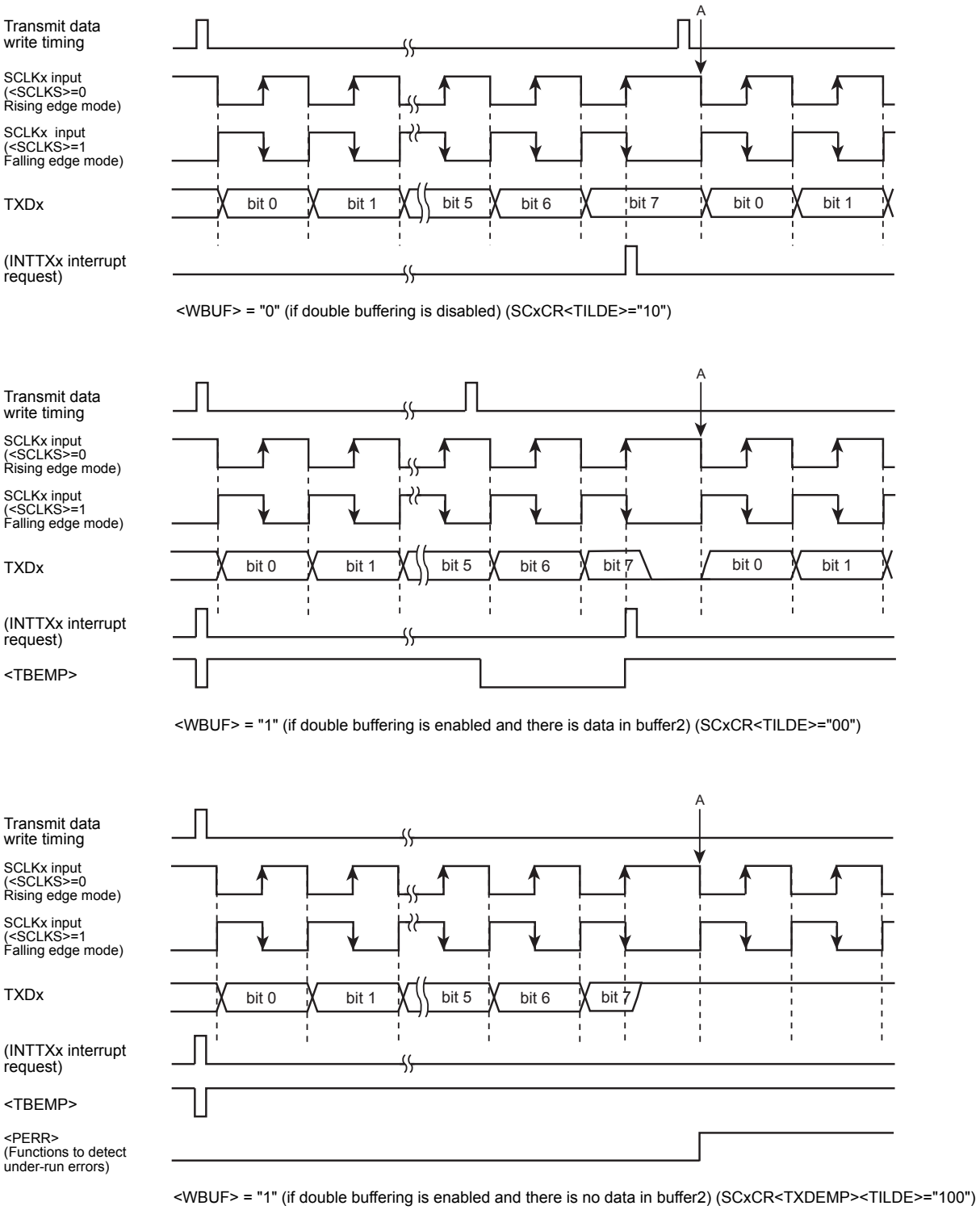


Figure 11-16 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

11.15.1.2 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 SCLKx pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data stored in the shift register is moved to the receive buffer and the shift register can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLK output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.

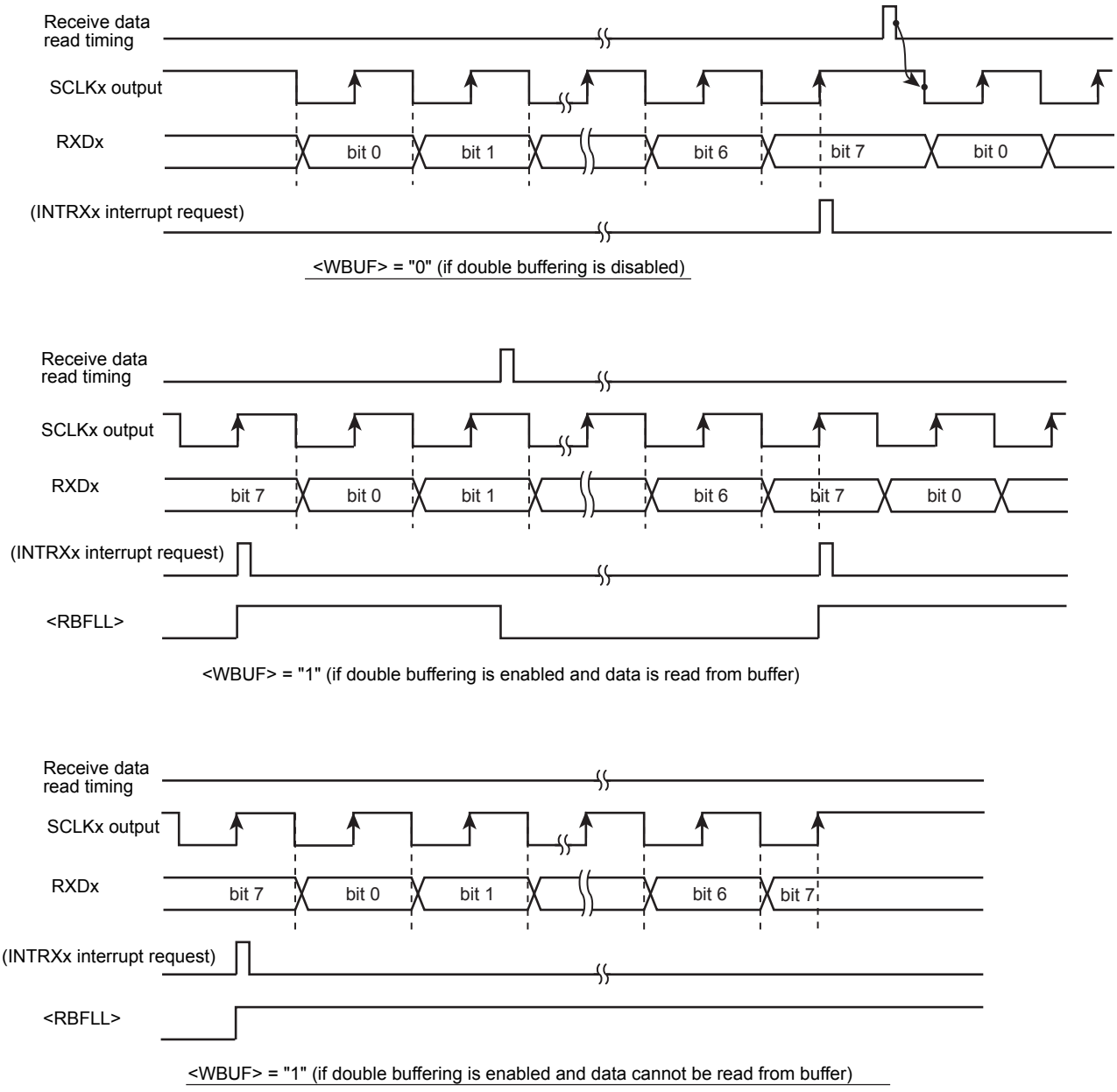


Figure 11-17 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to the receive buffer from the shift register, and the receive buffer can receive the next frame successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.

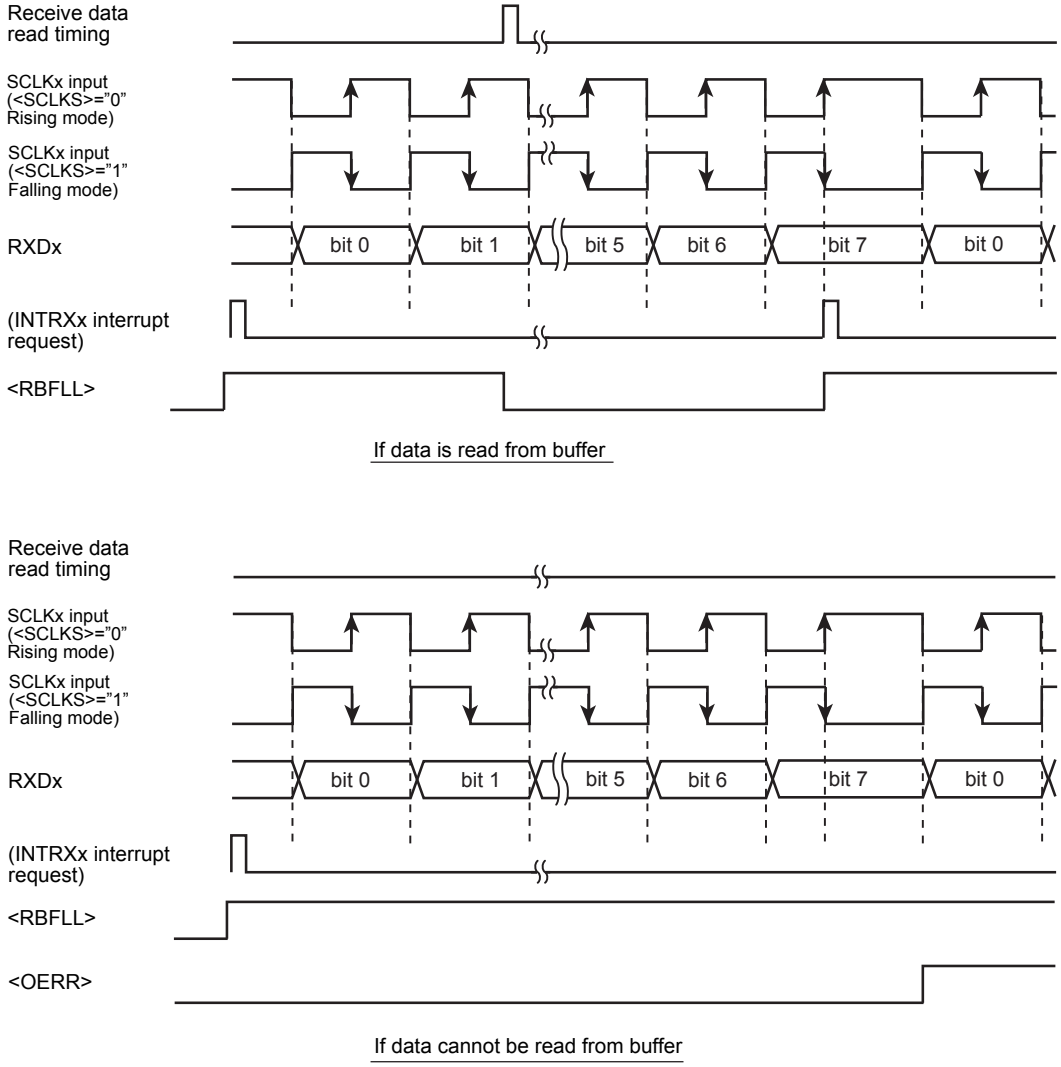


Figure 11-18 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

11.15.1.3 Transmit and Receive (Full-duplex)

(1) SCLK Output Mode

- If double buffers are disabled (SCxMOD2<WBUF> = "0")

  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 TXDx 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 double buffers are enabled (SCxMOD2<WBUF> = "1")

  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 TXDx pin. When all data bits are sent out, the INTTXx interrupt is generated and the next data is moved from the transmit buffer to the transmit shift register.

  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = "1") or when the receive buffer is full (SCxMOD2<RBFLL> = "1"), the SCLK output is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the next round of data transmission and reception is started.

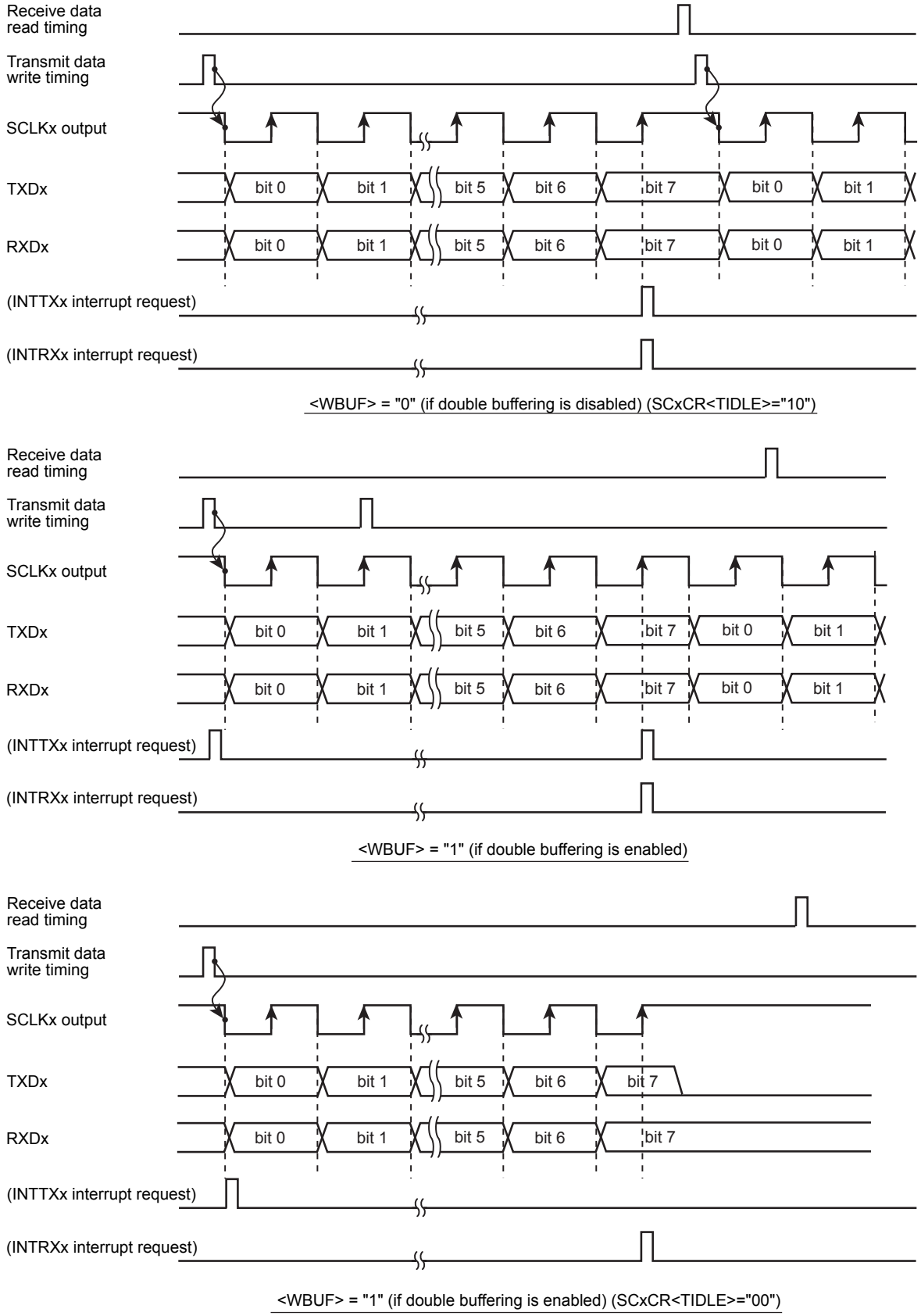


Figure 11-19 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If double buffers are disabled. (SCxMOD2<WBUF> = "0")

  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 TXDx pin and 8 bit of data is shifted into the receive buffer when the SCLK input becomes active. The INTTXx interrupt is generated upon completion of data transmission. The INTRXx interrupt is generated when the data is moved from 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 11-20). Data must be read before completing reception of the next frame data.

- If double buffers are enabled. (SCxMOD2<WBUF> = "1")

  The interrupt INTTXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx 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 11-20). 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 over-run error occurs.

  If there is no data written to transmit buffer when SCLK for the next frame is input, an under-run error occurs. The level which is specified by SCxCR<TXDEMP> is output to TXDx pin.

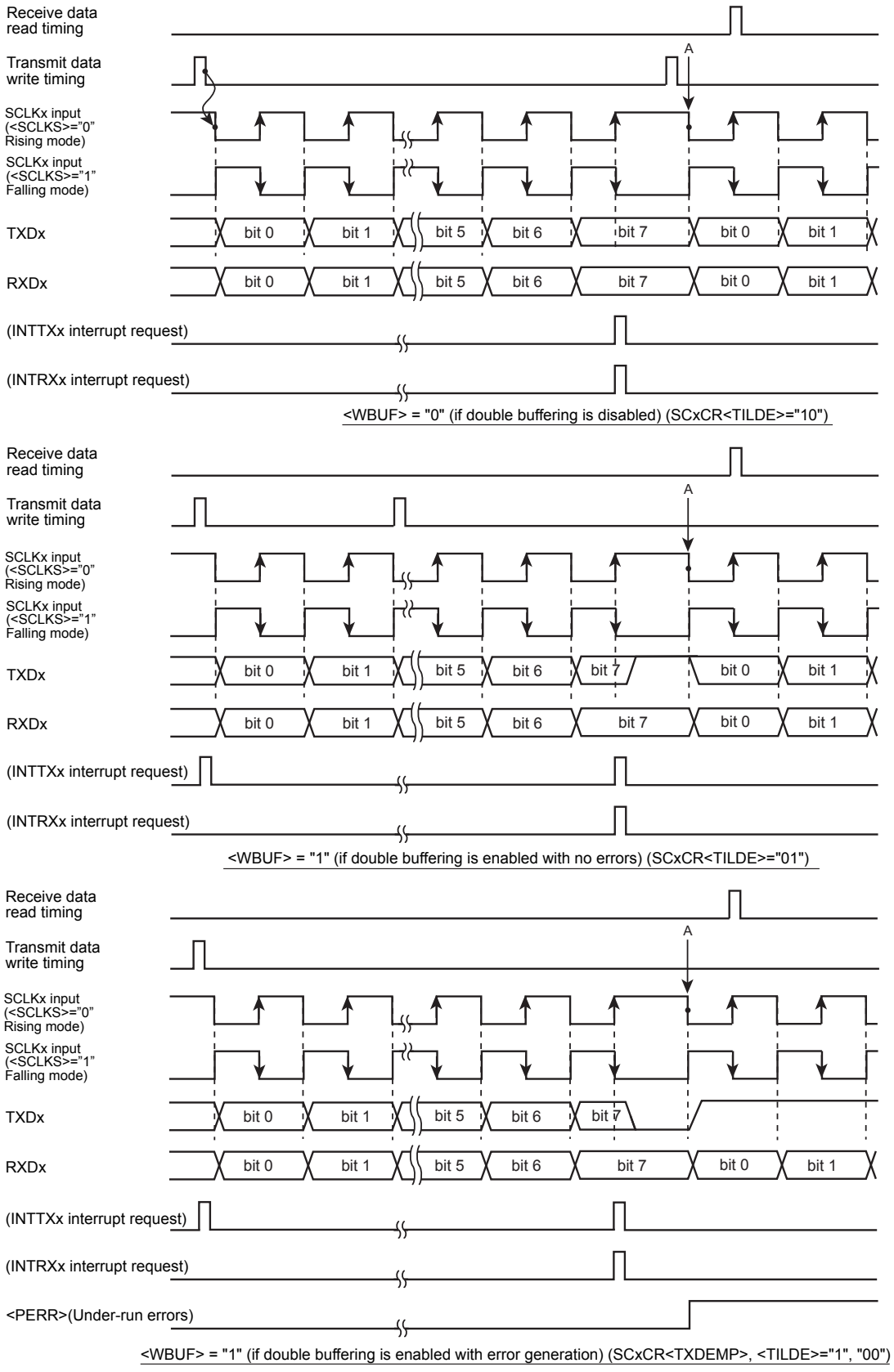


Figure 11-20 Transmit/Receive Operation in the I/O Interface Mode (SCLK Input Mode)

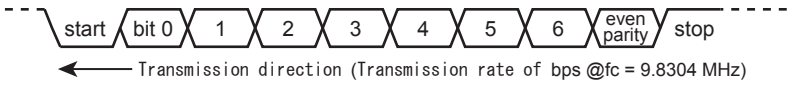
11.15.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SCxMOD0<SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SCxCR<PE>) controls the parity enable/disable setting.

When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN> bit. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.



Clocking conditions	system clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

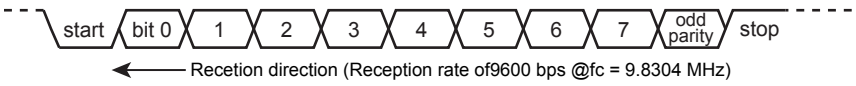
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x: don't care - : no change

11.15.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10". In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows:



Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	Set 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x: don't care - : no change

11.15.4 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 SCxMOD0<TB8> for transmitting data. The data is stored in SCxCR<RB8> for receiving data.

When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SCxBUF.

The stop bit length can be specified using SCxMOD2<SBLEN>.

11.15.4.1 Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SCxMOD0<WU> to "1".

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: The TXDx pin of the slave controller must be set to the open drain output mode using the PxOD register.

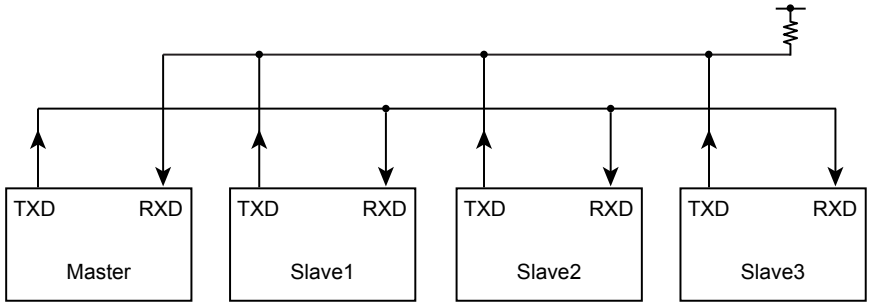
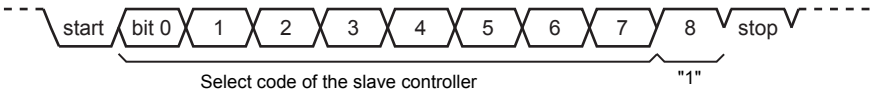


Figure 11-21 Serial Links to Use Wake-up Function

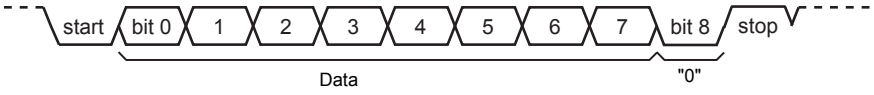
11.15.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.
2. Set SCxMOD0<WU> to "1" for the slave controllers to make them ready to receive data.
3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0".

5. The master controller transmits data to the designated slave controller (the controller of which <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.

12. Serial Bus Interface (I2C/SIO)

The TMPM3U0FSDMG contains 1 Serial Bus Interface (I2C/SIO) channel, in which the following two operating modes are included:

- I2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I2C bus mode, the I2C/SIO is connected to external devices via SCL and SDA.

In the clock-synchronous 8-bit SIO mode, the I2C/SIO is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the I2C/SIO in each operating mode.

Table 12-1 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
SBI	I2C bus mode	SCL:PB6 SDA:PF0	PBFR2[6]=1 PFFR2[0]=1	PBCR[6]=1 PFCR[0]=1	PBIE[6]=1 PFIE[0]=1	PBOD[6]=1 PFOD[0]=1
		SCL:PE2 SDA:PB5	PEFR5[2]=1 PBFR5[5]=1	PECR[2]=1 PBCR[5]=1	PEIE[2]=1 PBIE[5]=1	PEOD[2]=1 PBOD[5]=1
	SIO mode	SCK:PB5 SI:PB6 SO:PF0	PBFR3[5]=1 PBFR2[6]=1 PFFR2[0]=1	PBCR[6:5]=01(SCK0 output) PBCR[6:5]=00(SCK0 input) PFCR[0]=1	PBIE[6:5]=10(SCK0 output) PBIE[6:5]=11(SCL0 input) PFIE[0]=0	PBOD[6:5]=xx PFOD[0]=x

Note: x: Don't care

Note: In I2C bus mode , set a PB6&PF0 ports or PE2&PB5 ports .

12.1 Configuration

The configuration is shown in Figure 12-1.

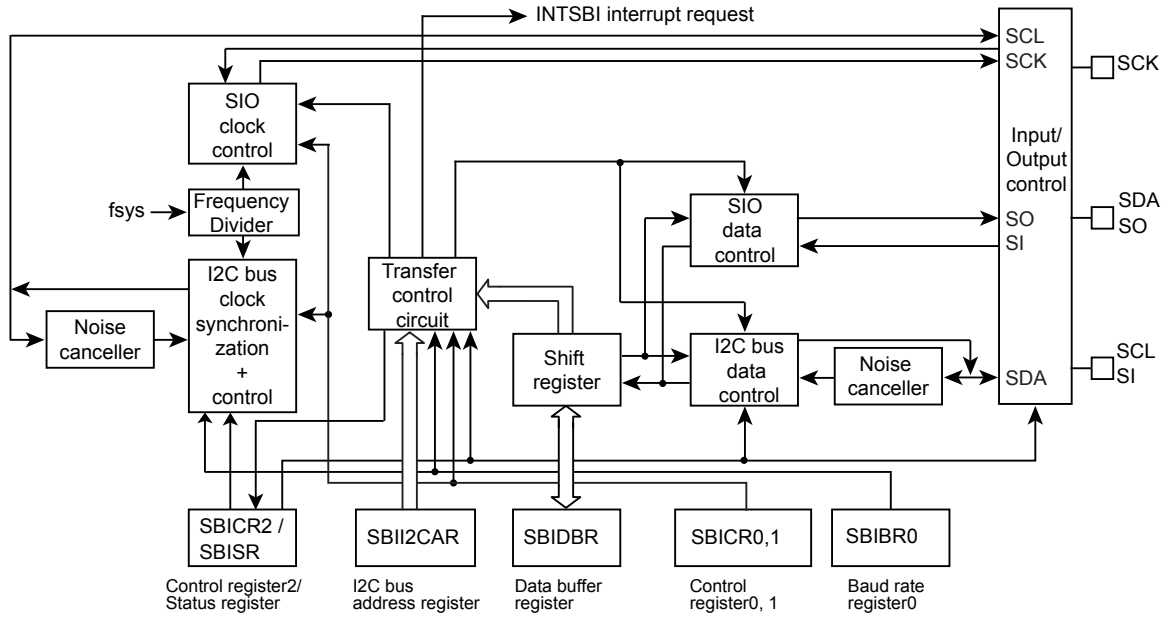


Figure 12-1 (I2C/SIO) Block Interface

12.2 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 "12.4 Control Registers in the I2C Bus Mode" and "12.8 Control register of SIO mode".

12.2.1 Registers for each channel

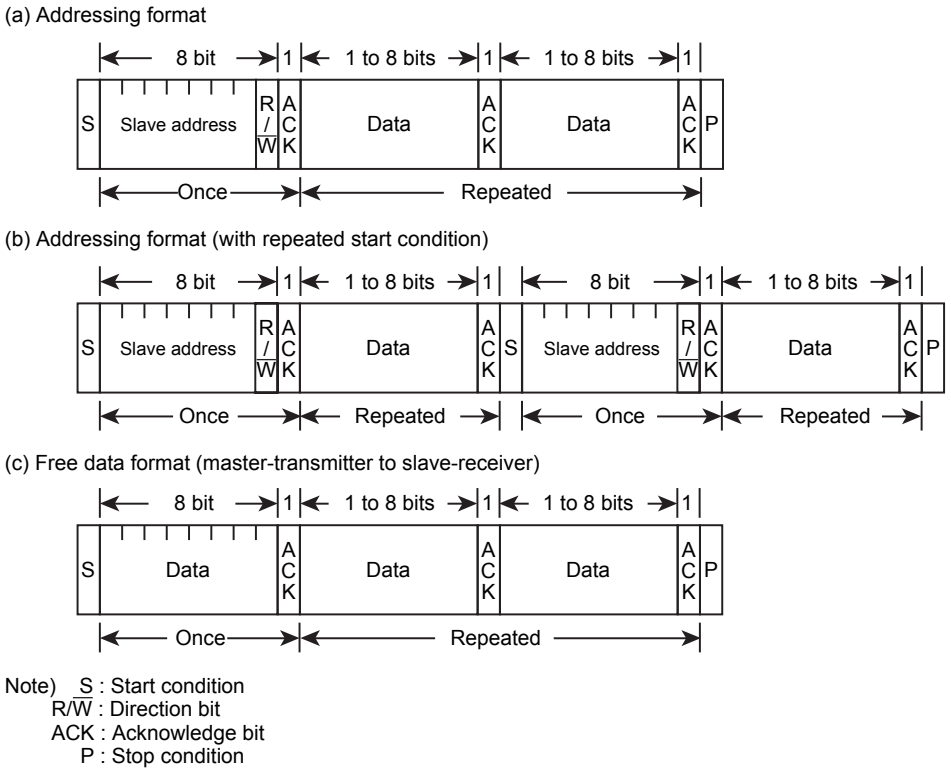
The tables below show the registers and register addresses for each channel.

Base Address = 0x4002_0000

Register name		Address(Base+)
Control register 0	SBICR0	0x0000
Control register 1	SBICR1	0x0004
Data buffer register	SBIDBR	0x0008
I2C bus address register	SBII2CAR	0x000C
Control register 2	SBICR2 (writing)	0x0010
Status register	SBISR (reading)	
Baud rate register 0	SBIBR0	0x0014

12.3 I2C Bus Mode Data Format

Figure 12-2 shows the data formats used in the I2C bus mode.



Note) S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 12-2 I2C Bus Mode Data Formats

12.4 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.

12.4.1 SBICR0(Control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBICR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it's been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

Note:To use the serial bus interface, enable this bit first.

12.4.2 SBICR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
After reset	0	0	0	0	1	0	0	1(Note3)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-5	BC[2:0]	R/W	Select the number of bits per transfer (Note 1)
4	ACK	R/W	Master mode 0: Acknowledgement clock pulse is not generated. 1: Acknowledgement clock pulse is generated. Slave mode 0: Acknowledgement clock pulse is not counted. 1: Acknowledgement clock pulse is counted.
3	-	R	Read as 1.
2-1	SCK[2:1]	R/W	Select internal SCL output clock frequency (Note 2).
0	SCK[0]	W	
	SWRMON	R	On reading <SWRMON>: Software reset status monitor 0:Software reset operation is in progress. 1:Software reset operation is not in progress.

<BC>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

000	n = 5	385 kHz
001	n = 6	294 kHz
010	n = 7	200 kHz
011	n = 8	122 kHz
100	n = 9	68 kHz
101	n = 10	36 kHz
110	n = 11	19 kHz
111		reserved

System Clock: fsys (= 40MHz)
Clock gear : fc/1
Frequency = $\frac{fsys}{2^n + 72}$ [Hz]

Note 1: **Clear <BC[2:0]> to "000" before switching the operation mode to the SIO mode.**

Note 2: **For details on the SCL line clock frequency, refer to "12.5.1 Serial Clock".**

Note 3: **After a reset, the <SCK[0]/SWRMON> bit is read as "1". However, if the SIO mode is selected at the SBICR2 register, the initial value of the <SCK[0]> bit is "0".**

Note 4: **The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.**

Note 5: **When <BC[2:0]>="001" and <ACK>="0" in master mode, SCL line may be fixed to "L" by falling edge of SCL line after generation of STOP condition and the other master devices can not use the bus. In the case of bus which is connected with several master devices, the bumber of bits per transfer should be set equal or more than 2 before generation of STOP condition.**

12.4.3 SBICR2(Control register 2)

This register serves as SBISR register by reading it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBI interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. For details, refer to "12.5.16 Software Reset".

Note: **Make sure that modes are not changed during a communication session.Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.**

12.4.4 SBISR (Status Register)

This register serves as SBICR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBI interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: − 1:Detected
2	AAS	R	Slave address match detection 0: − 1: Detected (This bit is set when the general-call address is detected as well.)
1	ADO	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

12.4.5 SBIBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

12.4.6 SBIDBR (Serial bus interface data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R (Receive)/ W (Transmit)	Receive data / Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIBDR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

12.4.7 SBII2CAR (I2Cbus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: **Please set the bit 0 <ALS> of I2C bus address register SBII2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.**

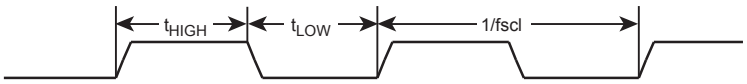
Note 2: **Do not set SBII2CAR to "0x00" in slave mode. (If SBII2CAR is set to "0x00", it's recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

12.5 Control in the I2C Bus Mode

12.5.1 Serial Clock

12.5.1.1 Clock source

SBICR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$t_{LOW} = 2^{n-1}/fsys + 58/fsys$
$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$
$fscl = 1/(t_{LOW} + t_{HIGH})$
$= \frac{fsys}{2^n + 72}$

SBICR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 12-3 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.**

12.5.1.2 Clock Synchronization

The I2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "Low" level overrides other masters producing the "High" level on their clock lines. This must be detected and responded by the masters producing the "High" level.

Clock synchronization assures correct data transfer on a bus that has two or more master.

For example, the clock synchronization procedure for a bus with two masters is shown below.

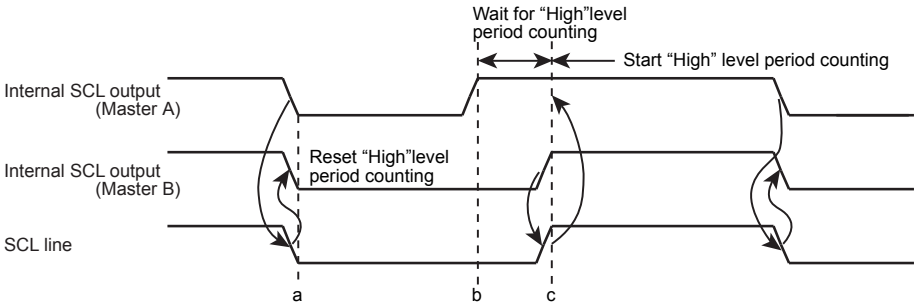


Figure 12-4 Example of Clock Synchronization

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.

12.5.2 Setting the Acknowledgement Mode

Setting SBICR1<ACK> to "1" selects the acknowledge mode.When operating as a master, the SBI adds one clock for acknowledgment signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the SBI releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. However, the second byte of the general call is necessary to be controlled by software to generate an acknowledgement signal depending on the contents of the second byte.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is counted.

12.5.3 Setting the Number of Bits per Transfer

SBICR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

12.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBII2CAR<ALS> and a slave address in SBII2CAR<SA[6:0]> sets addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

12.5.5 Operating mode

The setting of SBICR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, ensure that the serial bus interface pins are at "High" level before setting <SBIM[1:0]> to "10". Also, ensure that the bus is free before switching the operating mode to the port mode.

12.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBICR2<TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode:

- when data is transmitted in the addressing format.
- when the received slave address matches the value specified at SBII2CAR.
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros.

If the value of the direction bit (R/$\overline{W}$) 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.

12.5.7 Configuring the SBI as a Master or a Slave

Setting SBICR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

12.5.8 Generating Start and Stop Conditions

When SBISR<BB> is "0", writing "1" to SBICR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

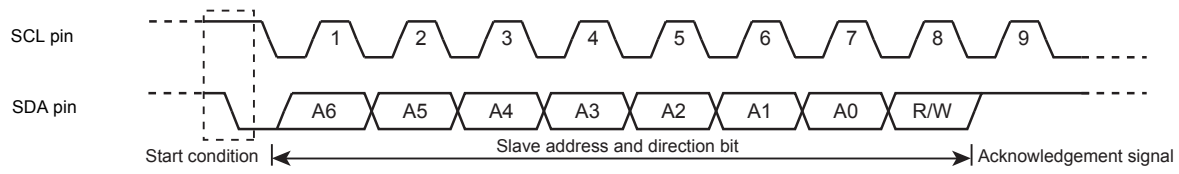


Figure 12-5 Generating the Start Condition and a Slave Address

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.

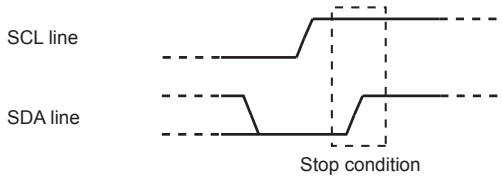


Figure 12-6 Generating the Stop Condition

SBISR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and cleared to "0" when the stop condition is detected (the bus is free).

12.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBI) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBI is generated under the following conditions.

- After output of the acknowledge signal which is generated when the received slave address matches the slave address set to SBII2CAR<SA[6:0]>.
- After the acknowledge signal is generated when a general-call address is received.
- When the slave address matches or a data transfer is completed after receiving a general-call address.

In the address recognition mode (<ALS> = "0"), INTSBI is generated when the received slave address matches the values specified at SBII2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBI) is generated, SBICR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the SBI pulls the SCL line to the "Low" level.

<PIN> is set to "1" when data is written to or read from SBIDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1". When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note: When When arbitration occurs while a slave address and direction bit are transferred in the master mode, <PIN> is cleared to "0" and INTSBI occurs. This does not relate to whether a slave address matches <SA>.

12.5.10 Arbitration Lost Detection Monitor

The I2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines.The I2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below.

Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the "Low" level and Master B outputs the "High" level.

Then Master A pulls the SDA bus line to the "Low" level because the line has the wired-AND connection. When the SCL line goes high at the point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid.

This condition of Master B is called "Arbitration Lost". Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

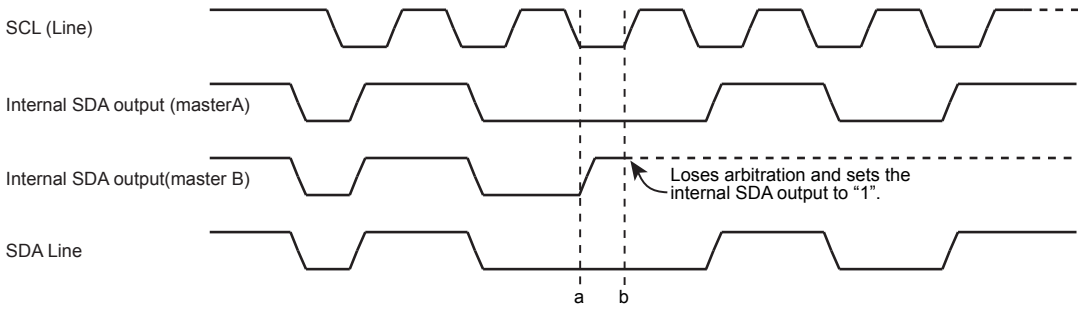


Figure 12-7 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs and SBISR<AL> is set to "1".

When an arbitration lost occures, SBISR<MST> and <TRX> are cleared to "0", causing the SBI to operate as a slave receiver and it stops the clock output during data transfer.If the master device which sends a slave address and direction bit generates Arbitration lost, it receives a slave address and direction bit which are sent by other master devices as slave device.Regardless of whether a received slave address matches <SA>, <PIN> is cleared to "0" and INTSBI occurs.

<AL> is cleared to "0" when data is written to or read from SBIDBR or data is written to SBICR2.

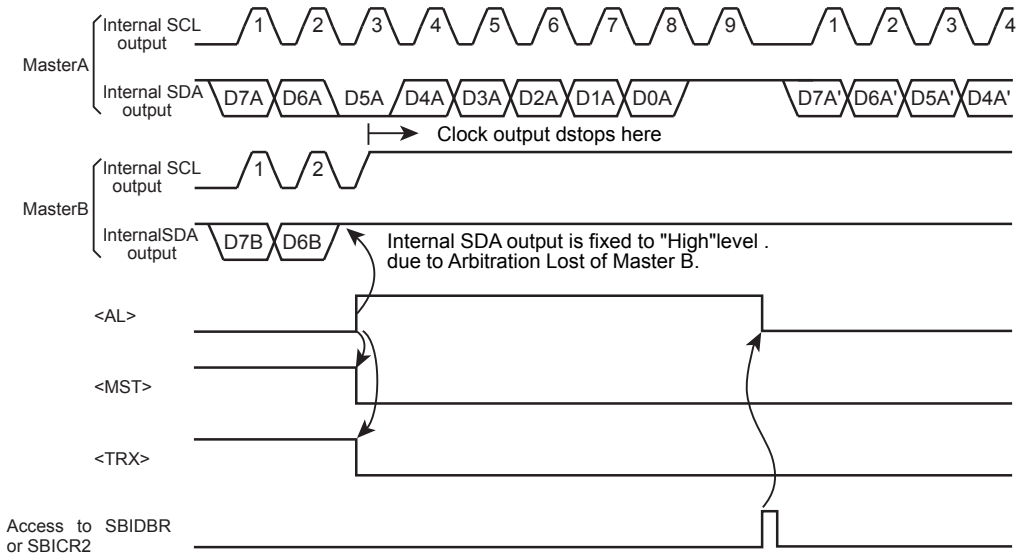


Figure 12-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

12.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBII2CAR<ALS>="0"), SBISR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBII2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIDBR.

12.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBISR<ADO> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<ADO> is cleared to "0" when the start or stop condition is detected on the bus.

12.5.13 Last Received Bit Monitor

SBISR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBISR<LRB> immediately after generation of the INTSBI interrupt request causes ACK signal to be read.

12.5.14 Data Buffer Register (SBIDBR)

Reading or writing SBIDBR initiates reading received data or writing transmitted data.

When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

12.5.15 Baud Rate Register (SBIBR0)

The SBIBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

12.5.16 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBICR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. When writing SBICR2<SWRST[1:0]>, set SBICR2<MST><TRX><BB><PIN> to "0000" and SBICR2<SBIM[1:0]> to "10" for I2C bus mode. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST[1:0]> is automatically cleared to "0".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

12.6 Data Transfer Procedure in the I2C Bus Mode

12.6.1 Device Initialization

Firstly, set SBICR1<ACK><SCK[2:0]>. Set "1" to <ACK> to specify the acknowledgement mode. Set "000" to SBICR1<BC[2:0]> .

Secondly, set <SA[6:0]> (a slave address) and <ALS> to SBII2CAR . (In the addressing format mode, set <ALS>="0").

Finally, to configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "000" to SBICR2<MST><TRX><BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "00" to <SWRST[1:0]>.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBICR1	←	0	0	0	1	0	X	X	X	Specifies ACK and SCL clock.
SBII2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBICR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don't care

12.6.2 Generating the Start Condition and a Slave Address

12.6.2.1 Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBICR1<ACK> to select the acknowledgment mode. Write to SBIDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBICR2<MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the master mode, the SBI holds the SCL line at the "Low" level while <PIN> is = "0".<TRX> changes its value according to the transmitted direction bit at generation of the INTSBI interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

```
                          7  6  5  4  3  2  1  0
Reg.          ←  SBISR
Reg.          ←  Reg. e 0x20
if Reg.       ≠  0x00                                  Ensures that the bus is free.

Then
SBICR1        ←  X  X  X  1  0  X  X  X   Selects the acknowledgement mode.
SBIDBR        ←  X  X  X  X  X  X  X  X   Specifies the desired slave address and direction.
SBICR2        ←  1  1  1  1  1  0  0  0   Generates the start condition.
Example of INTSBI0 interrupt routine
    Clears the interrupt request.
    Processing
    End of interrupt
```

12.6.2.2 Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line.

If the received address matches its slave address specified at SBII2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "Low" level during the ninth clock and outputs an acknowledgment signal.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the slave mode, the SBI holds the SCL line at the "Low" level while <PIN> is "0".

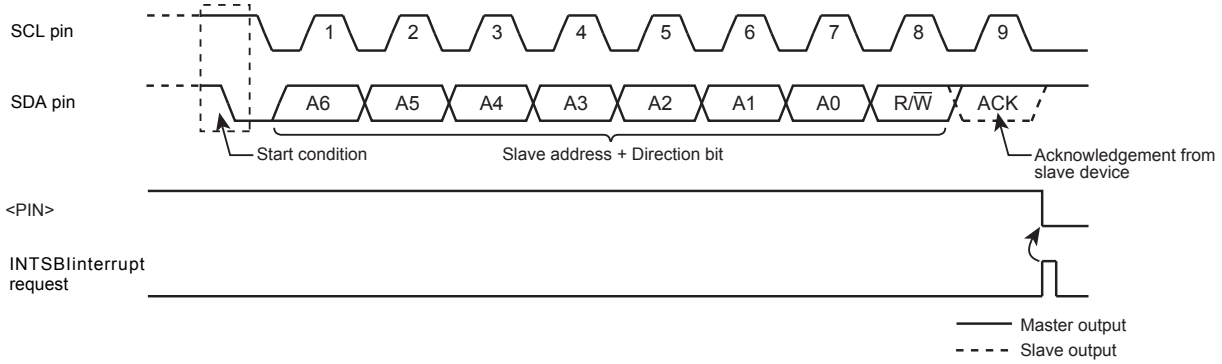


Figure 12-9 Generation of the Start Condition and a Slave Address

12.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBI interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

12.6.3.1 Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(1) Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1", that means the receiver requires no further data.

The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0", that means the receiver requires further data.If the next data to be transmitted has eight bits, the data is written into SBIDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIDBR.Writing the data makes <PIN> to "1", causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word.

After the transfer is completed, the INTSBI interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBI interrupt

if MST = 0

Then go to the slave-mode processing.

if TRX = 0

Then go to the receiver-mode processing.

if LRB = 0

Then go to processing for generating the stop condition.

SBICR1	←	X	X	X	X	0	X	X	X	Specifies the number of bits to be transmitted and specify whether ACK is required.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

End of interrupt processing.

Note: **X; Don't care**

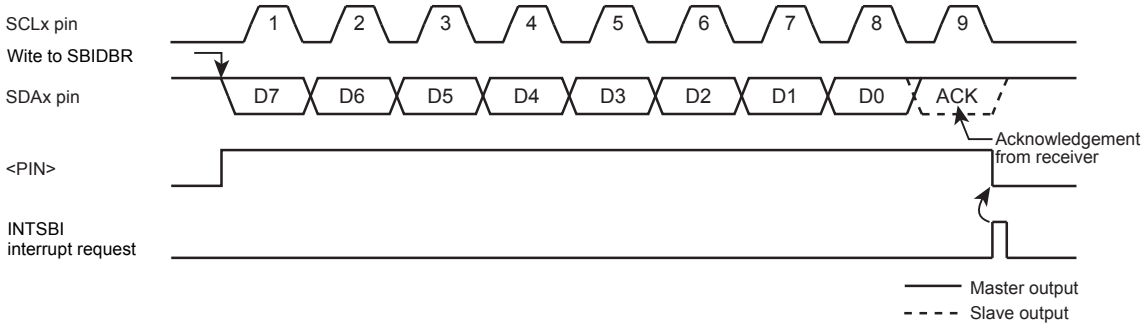


Figure 12-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIDBR.

If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)On reading the data, <PIN> is set to "1", and the serial clock is output to the SCL pin to transfer the next data word.In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SDA pin.

After that, the INTSBI interrupt request is generated, and <PIN> is cleared to "0", pulling the SCL pin to the "Low" level.Each time the received data is read from SBIDBR, one-word transfer clock and an acknowledgement signal are output.

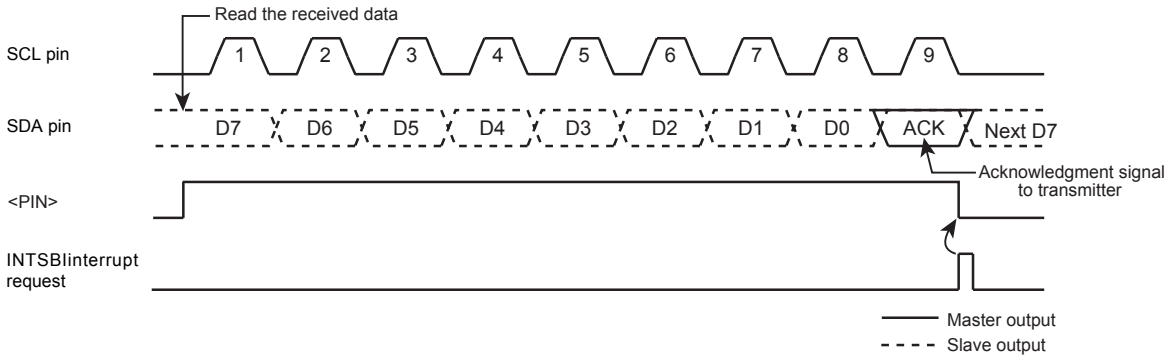


Figure 12-11 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be cleared to "0" immediately before reading the data word second to last.

This disables generation of an acknowledgment clock for the last data word.

When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC[2:0]> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer.

At this time, the master receiver holds the SDA bus line at the "High" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

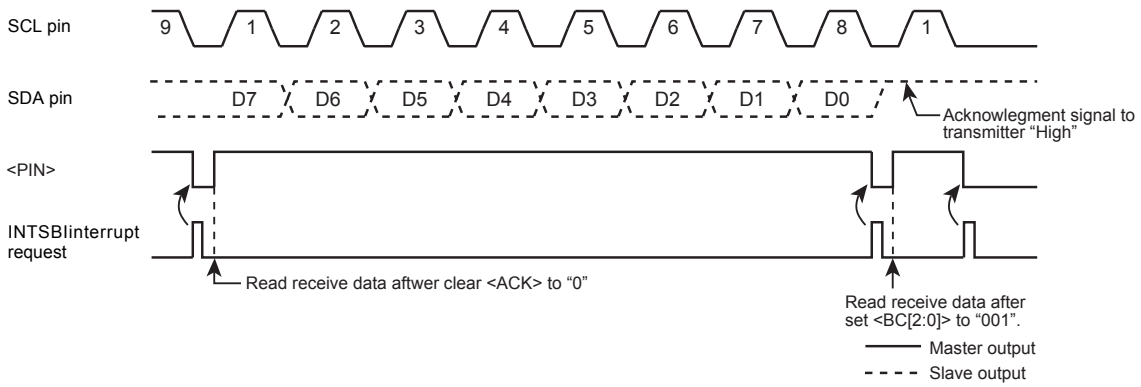


Figure 12-12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBI interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBICR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIDBR								Reads dummy data.
End of interrupt										

INTSBI interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIDBR								Reads the first to (N-2)th data words.
End of interrupt										

INTSBI interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBICR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIDBR								Reads the (N-1)th data word.
End of interrupt										

INTSBI interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBICR1	←	0	0	1	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIDBR								Reads the Nth data word.
End of interrupt										

INTSBI interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.
End of interrupt	

Note: X; Don't care

12.6.3.2 Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBI interrupt request on four occasions:

1) when the SBI has received any slave address from the master.

2) when the SBI has received a general-call address.

3) when the received slave address matches its address.

4) when a data transfer has been completed in response to a general-call.

Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode.

Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBI interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

When data is written to or read from SBIDBR or when <PIN> is set to "1", the SCLx pin is released after a period of t_{LOW}.

However, the second byte of the general call is necessary to be controlled by software to generate an acknowledgement signal depending on the contents of the second byte.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBISR<AL>, <TRX>, <AAS> and <ADO> are tested to determine the processing required.

"Table 12-2 Processing in Slave Mode"shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode.

INTSBI interrupt

```
if TRX = 0
Then go to other processing.
if AL = 0
Then go to other processing.
if AAS = 0
Then go to other processing.
SBICR1    ←  X  X  X  1  0  X  X  X    Sets the number of bits to be transmitted.
SBIDBR    ←  X  X  X  X  X  X  X  X    Sets the transmit data.
```

Note: X; Don't care

Table 12-2 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<ADO>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC[2:0]> and write the transmit data into SBIDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC[2:0].> and read the received data from SBIDBR.

12.6.4 Generating the Stop Condition

When SBISR<BB> is "1", writing "1" to SBICR2<MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus.

Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released.

After that, the SDA pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBICR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

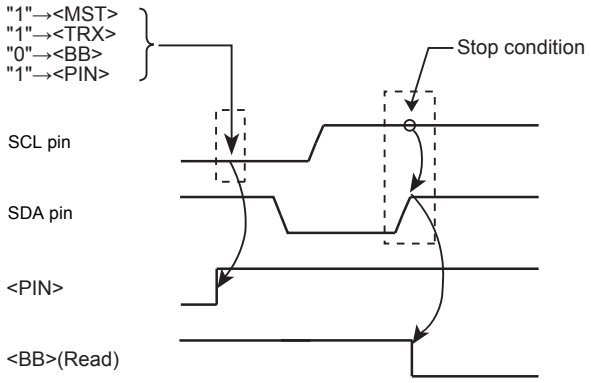


Figure 12-13 Generating the Stop Condition

12.6.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device.The procedure of generating a restart in the master mode is described below.

First, write SBICR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy.

Then, test SBISR<BB> and wait until it becomes "0" to ensure that the SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCLx bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "12.6.2 Generating the Start Condition and a Slave Address"to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

Note 1: **Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)**

Note 2: **When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a restart becomes "1", the rising edge of the SCL line is not detected even <LBR>=**

"1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.

		7	6	5	4	3	2	1	0	
SBICR2	←	0	0	0	1	1	0	0	0	Releases the bus.
if SBISR<BB> ≠ 0										Checks that the SCL pin is released.
Then										
if SBISR<LRB> ≠ 1										Checks that no other device is pulling the SCL pin to the "Low".
Then										
4.7 μs Wait										
SBICR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgment mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Sets the desired slave address and direction.
SBICR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Note: X; Don't care

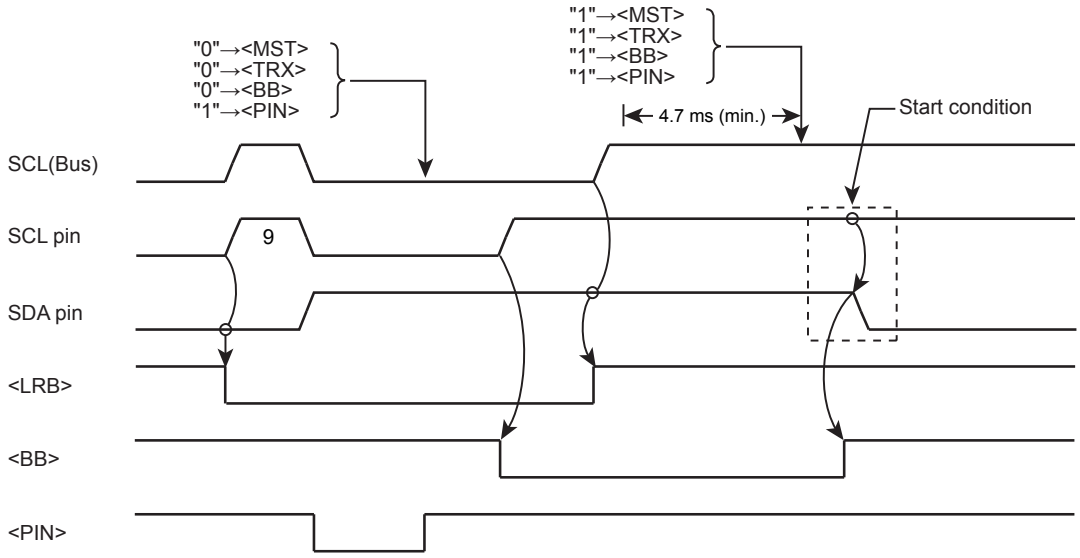


Figure 12-14 Timing Chart of Generating a Restart

12.7 Precautions on Use of Multi-master

Prepare recovery process by software in case that communication is in lock state in multi-master mode.

Example of recovery process

1. Start timer for timeout detection synchronizing with starting communication.
2. If a serial interface interrupt (INTSBI) does not occur within the specified time, a timeout occurs and the MCU determines that communication is locked up.
3. Do software reset on serial bus interface to release the condition that communication is locked up.
4. Adjust transmission timings. (Note)
5. Resend transmission data.

Note: Adjust transmission timing between the MCUs to avoid overlapping the transmission timing.

12.8 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.

12.8.1 SBICR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBICR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

12.8.2 SBICR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1)

000	n = 3	2.5 MHz
001	n = 4	1.25 MHz
010	n = 5	625 kHz
011	n = 6	313 kHz
100	n = 7	156 kHz
101	n = 8	78 kHz
110	n = 9	39 kHz
111	–	External clock

System clock: fsys (= 40MHz)
Clock gear: fc/1
Frequency = $\frac{fsys/2}{2^n}$ [Hz]

Note 1: **After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBICR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBICR2 register and the SBISR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

12.8.3 SBIDBR (Data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

12.8.4 SBICR2(Control register 2)

This register serves as SBISR register by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

Note 2: **Make sure that modes are not changed during a communication session.**

12.8.5 SBISR (Status Register)

This register serves as SBICR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	-	R	Read as 1.(Note)
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)

Note: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

12.8.6 SBIBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

12.9 Control in SIO mode

12.9.1 Serial Clock

12.9.1.1 Clock source

Internal or external clocks can be selected by programming SBICR1<SCK[2:0]>.

(1) Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

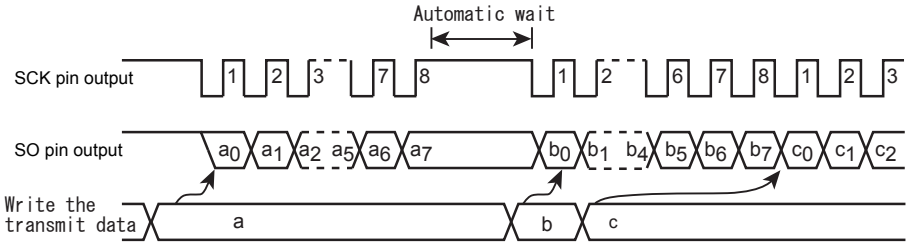


Figure 12-15 Automatic Wait

(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.

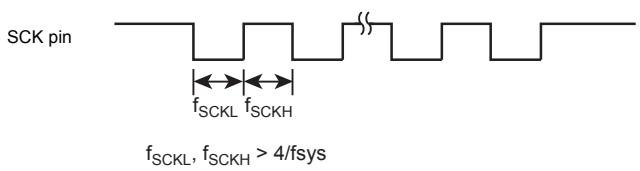


Figure 12-16 Maximum Transfer Frequency of External Clock Input

12.9.1.2 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).

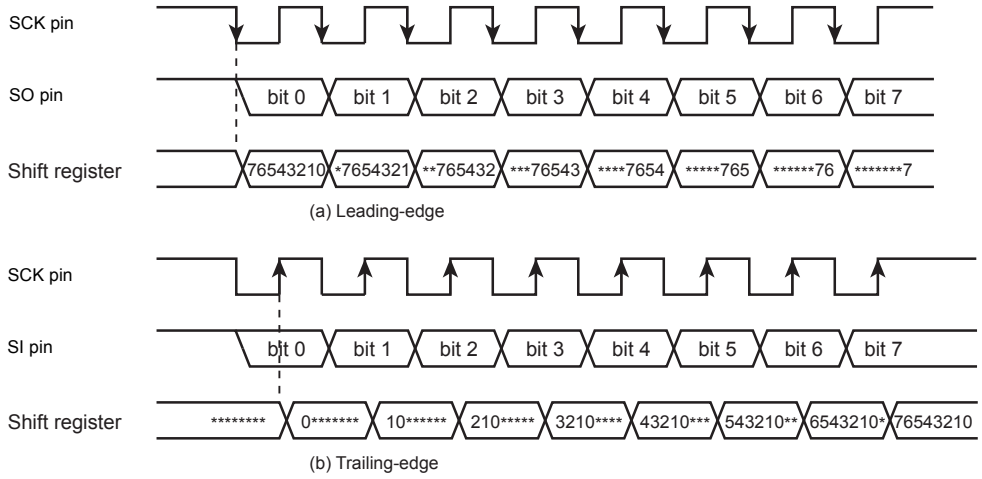


Figure 12-17 Shift Edge

12.9.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBICR1<SIOM[1:0]>.

12.9.2.1 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIDBR.

After writing the transmit data, writing "1" to SBICR1<SIOS> starts the transmission. The transmit data is moved from SBIDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIDBR becomes empty, and the INTSBI (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIDBR is loaded with the next transmit data.

In the external clock mode, SBIDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBISR<SIOF> to "1" to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBISR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBICR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBI interrupt

SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

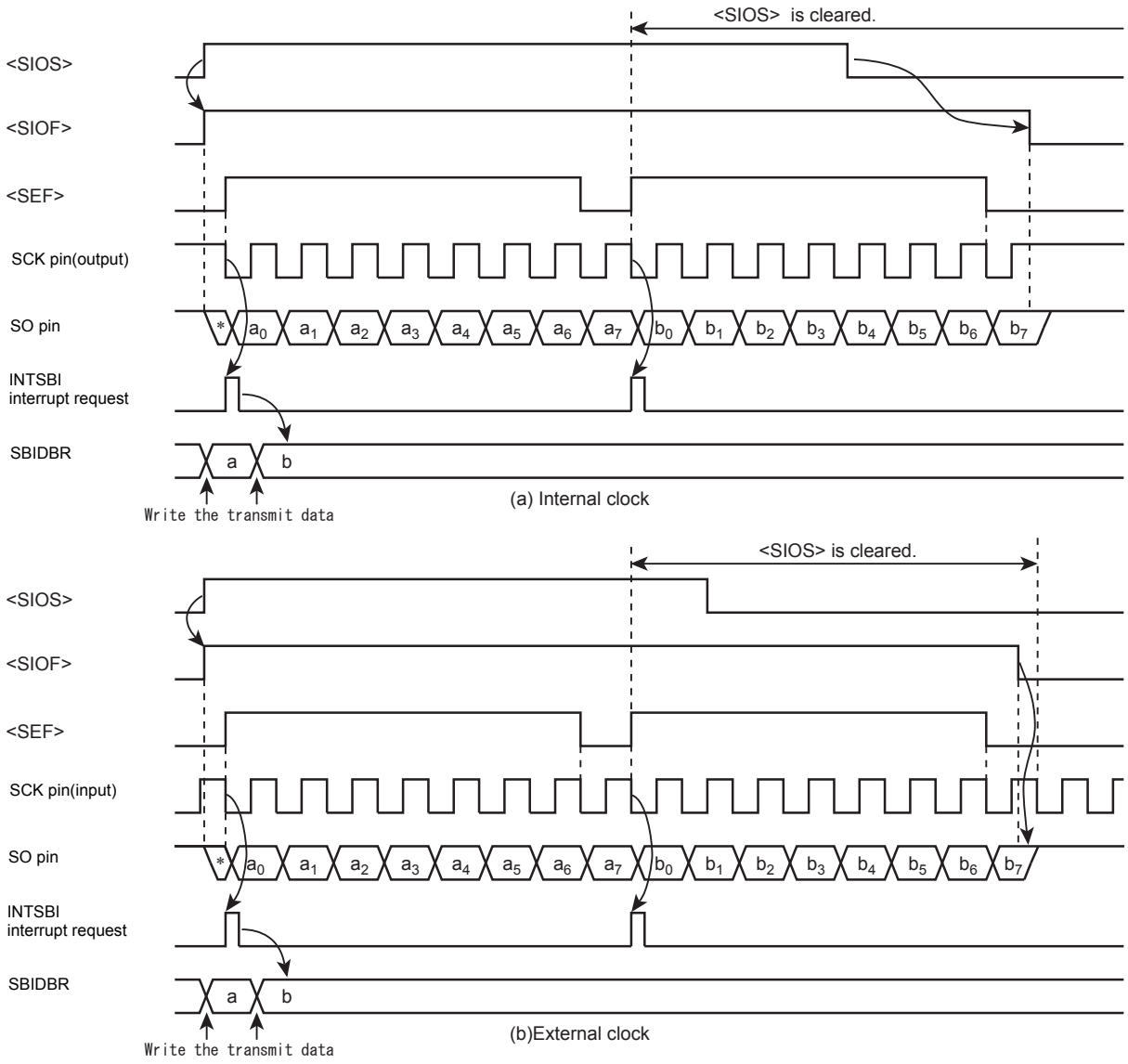


Figure 12-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

```
                                7  6  5  4  3  2  1  0
if SBISR<SIOF> ≠ 0                                        Recognizes the completion of the transmission.
Then
if SCK ≠ 1                                                Recognizes "1" is set to the SCK pin by monitoring the port.
Then
SBICR1          ←  0  0  0  0  0  1  1  1                 Completes the transmission by setting <SIOS> = 0.
```

12.9.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBICR1<SIOS> enables reception.Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIDBR and the INTSBI (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIDBR. The program checks SBISR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: **The contents of SBIDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBICR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBI interrupt

Reg.	←	SBIDBR	Reads the received data.

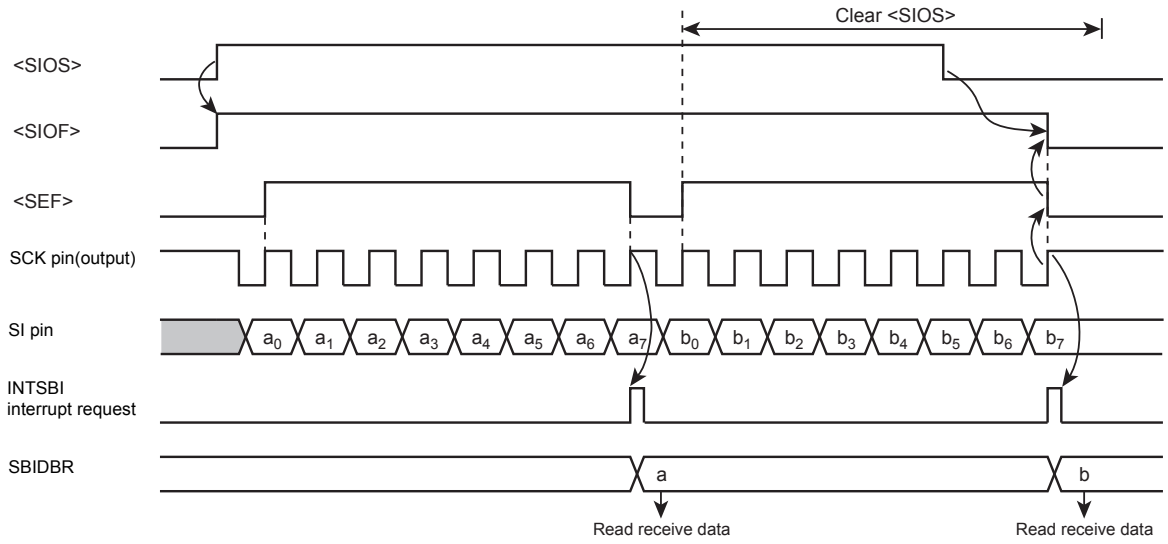


Figure 12-19 Receive Mode (Example: Internal Clock)

12.9.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIDBR and setting SBICR1<SIOS> to "1" enables transmission and reception.The transmit data is output through the SOx pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIDBR and the INTSBI interrupt request is generated.The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started.The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBICR1<SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIDBR. The program checks SBISR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception.If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: **The contents of SBIDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

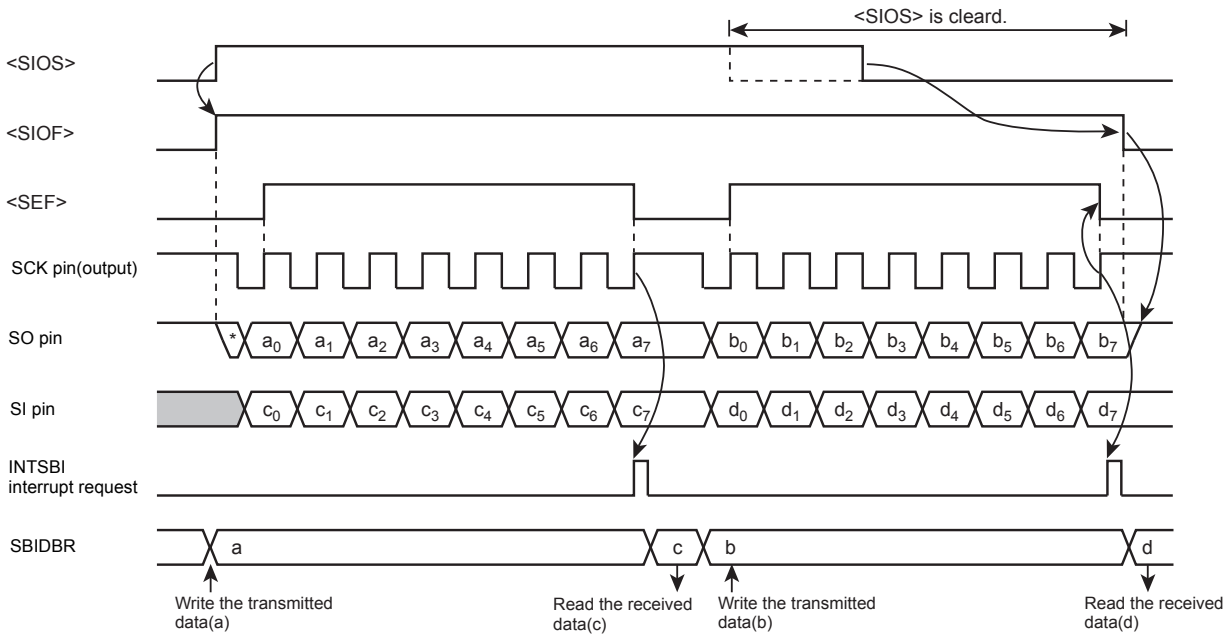


Figure 12-20 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBICR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBI interrupt

Reg.	←	SBIDBR								Reads the received data.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

12.9.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBICR1<SIOS>= "0", the last bit of the transmitted data retains the data of SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

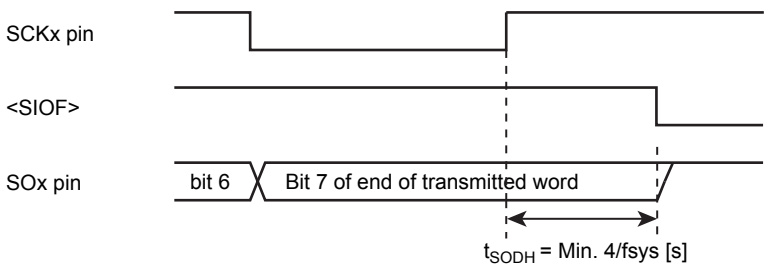


Figure 12-21 Data retention time of the last bit at the end of transmission

13. 12-Bit Analog-to-Digital Converter

The TMPM3U0FSDMG contains a 12-bit successive-approximation analog-to-digital converter (ADC). The ADC unit B (ADC B) has 4 analog inputs. The ADC cooperates with the PMD circuit to support motor control.

Four external analog input pins (AINB9 to AINB12) can also be used as input/output ports.

13.1 Functions and features

1. It can select analog input and start AD conversion when receiving trigger signal from PMD or TMRB(interrupt).
2. It can select analog input, in the Software Trigger Program and the Constant Trigger Program.
3. The ADCs has twelve register for AD conversion result.
4. The ADCs generate interrupt signal at the end of the program which was started by PMD trigger and TMRB trigger.
5. The ADCs generate interrupt signal at the end of the program which are the Software Trigger Program and the Constant Trigger Program.
6. The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

13.2 Block Diagram

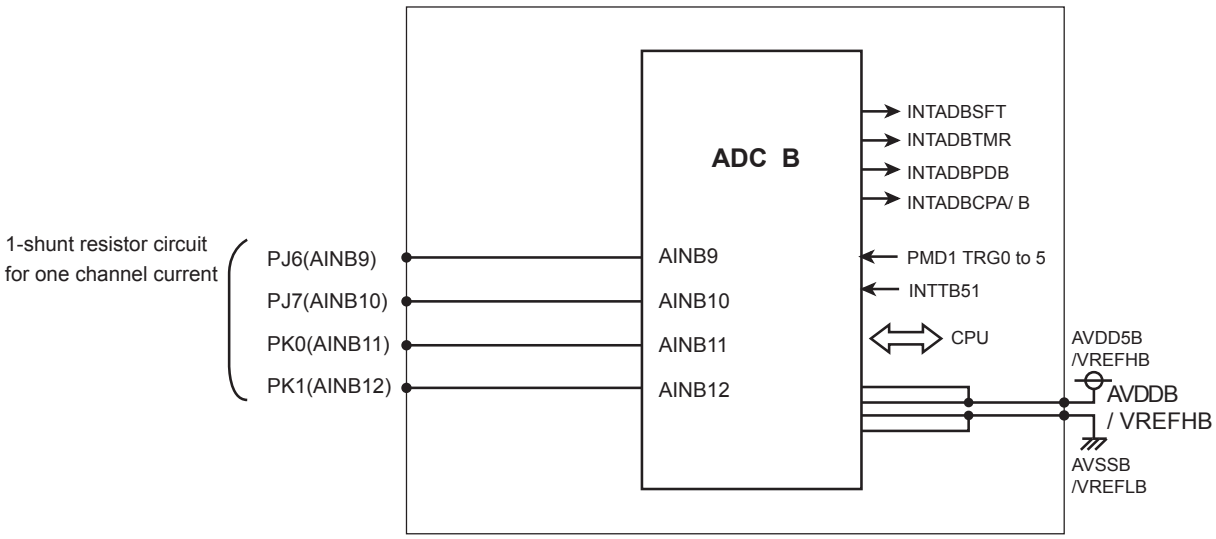


Figure 13-1 AD converters Block Diagram

13.3 List of Registers

Unit	Base Address
Unit B	0x4003_0200

Register Name(x=B)		Address(Base+)
Clock Setting Register	ADxCLK	0x0000
Mode Setting Register 0	ADxMOD0	0x0004
Mode Setting Register 1	ADxMOD1	0x0008
Mode Setting Register 2	ADxMOD2	0x000C
Monitoring Setting Register 0	ADxCMPCR0	0x0010
Monitoring Setting Register 1	ADxCMPCR1	0x0014
Conversion Result Compare Register 0	ADxCMP0	0x0018
Conversion Result Compare Register 1	ADxCMP1	0x001C
Conversion Result Register 0	ADxREG0	0x0020
Conversion Result Register 1	ADxREG1	0x0024
Conversion Result Register 2	ADxREG2	0x0028
Conversion Result Register 3	ADxREG3	0x002C
Conversion Result Register 4	ADxREG4	0x0030
Conversion Result Register 5	ADxREG5	0x0034
Conversion Result Register 6	ADxREG6	0x0038
Conversion Result Register 7	ADxREG7	0x003C
Conversion Result Register 8	ADxREG8	0x0040
Conversion Result Register 9	ADxREG9	0x0044
Conversion Result Register 10	ADxREG10	0x0048
Conversion Result Register 11	ADxREG11	0x004C
-	Reserved	0x0050
-	Reserved	0x0054
-	Reserved	0x0058
-	Reserved	0x005C
-	Reserved	0x0060
-	Reserved	0x0064
PMD Trigger Program Number Select Register 6	ADxPSEL6	0x0068
PMD Trigger Program Number Select Register 7	ADxPSEL7	0x006C
PMD Trigger Program Number Select Register 8	ADxPSEL8	0x0070
PMD Trigger Program Number Select Register 9	ADxPSEL9	0x0074
PMD Trigger Program Number Select Register 10	ADxPSEL10	0x0078
PMD Trigger Program Number Select Register 11	ADxPSEL11	0x007C
PMD Trigger Interrupt Select Register 0	ADxPINTS0	0x0080
PMD Trigger Interrupt Select Register 1	ADxPINTS1	0x0084
PMD Trigger Interrupt Select Register 2	ADxPINTS2	0x0088
PMD Trigger Interrupt Select Register 3	ADxPINTS3	0x008C
PMD Trigger Interrupt Select Register 4	ADxPINTS4	0x0090
PMD Trigger Interrupt Select Register 5	ADxPINTS5	0x0094
PMD Trigger Program Register 0	ADxPSET0	0x0098
PMD Trigger Program Register 1	ADxPSET1	0x009C
PMD Trigger Program Register 2	ADxPSET2	0x00A0
PMD Trigger Program Register 3	ADxPSET3	0x00A4
PMD Trigger Program Register 4	ADxPSET4	0x00A8
PMD Trigger Program Register 5	ADxPSET5	0x00AC
Timer Trigger Program Registers 0 to 3	ADxTSET03	0x00B0

Register Name(x=B)		Address(Base+)
Timer Trigger Program Registers 4 to 7	ADxTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADxTSET811	0x00B8
Software Trigger Program Registers 0 to 3	ADxSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADxSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADxSSET811	0x00C4
Constant Conversion Program Registers0 to 3	ADxASET03	0x00C8
Constant Conversion Program Registers 4 to 7	ADxASET47	0x00CC
Constant Conversion Program Registers 8 to 11	ADxASET811	0x00D0
Mode Setting Register 3	ADxMOD3	0x00D4

Note: Access to the "Reserved" areas are prohibited.

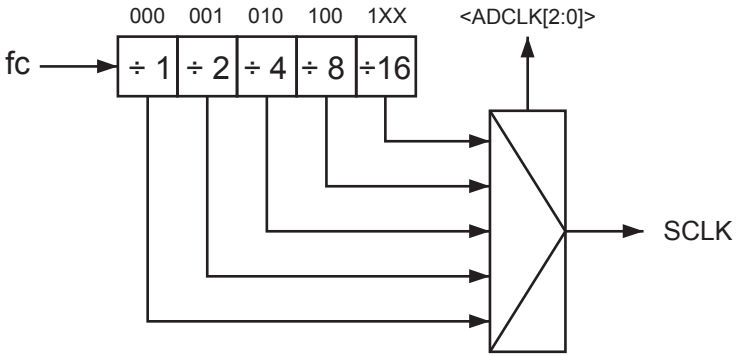
13.4 Register Descriptions

AD conversion is performed at the clock frequency selected in the ADC Clock Setting Register.

13.4.1 ADxCLK (Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-3	TSH[3:0]	R/W	Write as "1001".
2-0	ADCLK[2:0]	R/W	AD prescaler output (SCLK) select 000: fc (Note1) 001: fc/2 010: fc/4 011: fc/8 1xx: fc/16



Note 1: Frequency of SCLK can be use up to 40MHz.

Note 2: AD conversion is performed at the clock frequency selected in this register. The conversion clock frequency must be selected to ensure the guaranteed accuracy.

Note 3: The conversion clock must not be changed while AD conversion is in progress.

13.4.2 ADxMOD0 (Mode Setting Register 0)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	DACON	R/W	DAC control 0: OFF 1: ON Setting <DACON> to "1", when using the ADC.
0	ADSS	W	Software triggered conversion 0: Don't care 1: Start Setting <ADSS> to "1" starts AD conversion (software triggered conversion). Receiving trigger signal from PMD or TMRB(interrupt) starts AD conversion also. For detail setting, please read the chapter about PMD and TMRB.

13.4.3 ADxMOD1 (Mode Setting Register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	ADEN	R/W	AD conversion control 0: Disable 1: Enable Setting <ADEN> to "1", when using the ADC. After Setting <ADEN> to "1", setting <ADAS> to "1" starts AD conversion and repeat conversion.
6-1	−	R	Read as "0".
0	ADAS	R/W	Constant AD conversion control 0: Disable 1: Enable

13.4.4 ADxMOD2 (Mode Setting Register 2)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	ADSFN	R	Software conversion busy flag 0: Conversion completed 1: Conversion in progress The <ADSFN> is a software AD conversion busy flag. After <ADSS> was set to "1", when AD conversion is actually started, <ADSFN> is set to "1". When finished AD conversion, <ADSFN> is cleared to "0".
0	ADBFN	R	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress The <ADBFN> is an AD conversion busy flag. When AD conversion is started regardless of conversion factor (PMD, Timer, Software, Constant), <ADBFN> is set to "1". When finished AD conversion, <ADBFN> is cleared to "0".

13.4.5 ADxMOD3 (Mode Setting Register 3)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-11	−	R/W	Write as "0".
10	−	R/W	Write as "1".
9	−	R/W	Write as "0".
8	RCUT	R/W	ADC operational control 2 0: Enable 1: Disable When ADC is operated, write to "0" in advance. While ADC stops the operation, it can be reduced a power cunsumption by setting to "1".
7	−	R/W	Write as "0".
6	−	R/W	Write as "1".
5-3	PMODE[2:0]	R/W	Write as "100".
2-0	−	R/W	Write as "0".

Note: ADxMOD3<PMODE[2:0]> must be set to "100". And do not change other bits in ADxMOD3 register.

13.4.6 ADxCMPCR0(Monitoring Setting Register 0)

After fixing the conversion result, the interrupt signal (INTADxCPn) is generated.

(x=B,n=A ;A: Monitor0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT0[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons ・ ・ 15: After 16 comparisons The ADxCMPCR0 and ADxCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
7	CMP0EN	R/W	Monitoring function 0:Disable 1:Enable
6-5	-	R	Read as "0".
4	ADBIG0	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register
3-0	REGS0[3:0]	R/W	AD conversion result register to be compared 0000: ADxREG0 / 0001: ADxREG1 / 0010: ADxREG2 / 0011: ADxREG3 0100: ADxREG4 / 0101: ADxREG5 / 0110: ADxREG6 / 0111: ADxREG7 1000: ADxREG8 / 1001: ADxREG9 / 1010: ADxREG10 / 1011: ADxREG11

13.4.7 ADxCMPCR1(Monitoring Setting Register 1)

After fixing the conversion result, the interrupt signal (INTADxCPn) is generated.

(x=B,n=B ; B: Monitor1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT1[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons ・ ・ 15: After 16 comparisons The ADxCMPCR0 and ADxCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
7	CMP1EN	R/W	Monitoring function 0:Disable 1:Enable
6-5	-	R	Read as "0".
4	ADBIG1	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register
3-0	REGS1[3:0]	R/W	AD conversion result register to be compared 0000: ADxREG0 / 0100: ADxREG4 / 1000: ADxREG8 0001: ADxREG1 / 0101: ADxREG5 / 1001: ADxREG9 0010: ADxREG2 / 0110: ADxREG6 / 1010: ADxREG10 0011: ADxREG3 / 0111: ADxREG7 / 1011: ADxREG11

13.4.8 ADxCMP0(Conversion Result Compare Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP[11:0]	R/W	The value to be compared with an AD conversion result. Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

13.4.9 ADxCMP1(Conversion Result Compare Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD1CMP							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD1CMP				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD1CMP[11:0]	R/W	The value to be compared with an AD conversion result. Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

13.4.10 ADxREG0(Conversion Result Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR00							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR00				-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR00[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR0	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG0 is read and is cleared when the low-order byte of ADxREG0 is read.
0	ADR0RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR0RF> is a flag that is set when an AD conversion result is stored in the ADxREG0 register and is cleared when the low-order byte of ADxREG0 is read.

13.4.11 ADxREG1(Conversion Result Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR10[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR1	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG1 is read and is cleared when the low-order byte of ADxREG1 is read.
0	ADR1RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR1RF> is a flag that is set when an AD conversion result is stored in the ADxREG1 register and is cleared when the low-order byte of ADxREG1 is read.

13.4.12 ADxREG2(Conversion Result Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR20							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR20				-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR20[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR2	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG2 is read and is cleared when the low-order byte of ADxREG2 is read.
0	ADR2RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR2RF> is a flag that is set when an AD conversion result is stored in the ADxREG2 register and is cleared when the low-order byte of ADxREG2 is read.

13.4.13 ADxREG3(Conversion Result Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR30							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR30				-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR30[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR3	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG3 is read and is cleared when the low-order byte of ADxREG3 is read.
0	ADR3RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR3RF> is a flag that is set when an AD conversion result is stored in the ADxREG3 register and is cleared when the low-order byte of ADxREG3 is read.

13.4.14 ADxREG4(Conversion Result Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR40							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR40				-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR40[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR4	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG4 is read and is cleared when the low-order byte of ADxREG4 is read.
0	ADR4RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR4RF> is a flag that is set when an AD conversion result is stored in the ADxREG4 register and is cleared when the low-order byte of ADxREG4 is read.

13.4.15 ADxREG5(Conversion Result Register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR50							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR50				-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR50[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR5	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG5 is read and is cleared when the low-order byte of ADxREG5 is read.
0	ADR5RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR5RF> is a flag that is set when an AD conversion result is stored in the ADxREG5 register and is cleared when the low-order byte of ADxREG5 is read.

13.4.16 ADxREG6(Conversion Result Register 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR60							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR60				-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR60[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR6	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG6 is read and is cleared when the low-order byte of ADxREG6 is read.
0	ADR6RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR6RF> is a flag that is set when an AD conversion result is stored in the ADxREG6 register and is cleared when the low-order byte of ADxREG6 is read.

13.4.17 ADxREG7(Conversion Result Register 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR70							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR70				-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR70[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR7	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG7 is read and is cleared when the low-order byte of ADxREG7 is read.
0	ADR7RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR7RF> is a flag that is set when an AD conversion result is stored in the ADxREG7 register and is cleared when the low-order byte of ADxREG7 is read.

13.4.18 ADxREG8(Conversion Result Register 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR80							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR80				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR80[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR8	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG8 is read and is cleared when the low-order byte of ADxREG8 is read.
0	ADR8RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR8RF> is a flag that is set when an AD conversion result is stored in the ADxREG8 register and is cleared when the low-order byte of ADxREG8 is read.

13.4.19 ADxREG9(Conversion Result Register 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR90							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR90				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR90[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR9	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG9 is read and is cleared when the low-order byte of ADxREG9 is read.
0	ADR9RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR9RF> is a flag that is set when an AD conversion result is stored in the ADxREG9 register and is cleared when the low-order byte of ADxREG9 is read.

13.4.20 ADxREG10(Conversion Result Register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR100							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR100				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR100[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR10	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG10 is read and is cleared when the low-order byte of ADxREG10 is read.
0	ADR10RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR10RF> is a flag that is set when an AD conversion result is stored in the ADxREG10 register and is cleared when the low-order byte of ADxREG10 is read.

13.4.21 ADxREG11(Conversion Result Register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR110							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR110				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR110[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR11	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG11 is read and is cleared when the low-order byte of ADxREG11 is read.
0	ADR11RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR11RF> is a flag that is set when an AD conversion result is stored in the ADxREG11 register and is cleared when the low-order byte of ADxREG11 is read.

13.4.22 PMD Trigger Program Registers

AD conversion can be started by a trigger from the PMD (programmable motor driver).

The PMD trigger program registers are used to specify the program to be started by each of six triggers generated by the PMD, to select the interrupt to be generated upon completion of the program and to select the AIN input to be used.

The PMD trigger program registers include three types of registers.

(x=B : ADC Unit)

- PMD Trigger Program Number Select Register (ADxPSEL6 to ADxPSEL11)

  The PMD Trigger Program Number Select Register (ADxPSELn) specifies the program to be started by each of six AD conversion start signals corresponding to six triggers(PMD1TRG0 to 5) generated by the PMD. Programs 0 to 5 are available.

  "ADxPSEL6 to ADxPSEL11" corresponds to "PMD1TRG0 to 5".
- PMD Trigger Interrupt Select Register (ADxPINTS0 to ADxPINTS5)

  The PMD Trigger Interrupt Select Registers (ADxPINTS0 to ADxPINTS5) select the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

  ADxPINTS0 corresponds to program 0, and it exists to ADxPINT5 (program 5).
- PMD Trigger Program Register (ADxPSET0 to ADxPSET5)

  The PMD Trigger Program Setting Registers (ADxPSET0 to ADxPSET5) specify the settings for each of programs 0 to 5. Each PMD Trigger Program Register is comprised of four registers for specifying the AIN input to be converted. The conversion results corresponding to the ADxPSETn0 to ADxPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADxREG0 to ADxREG3).

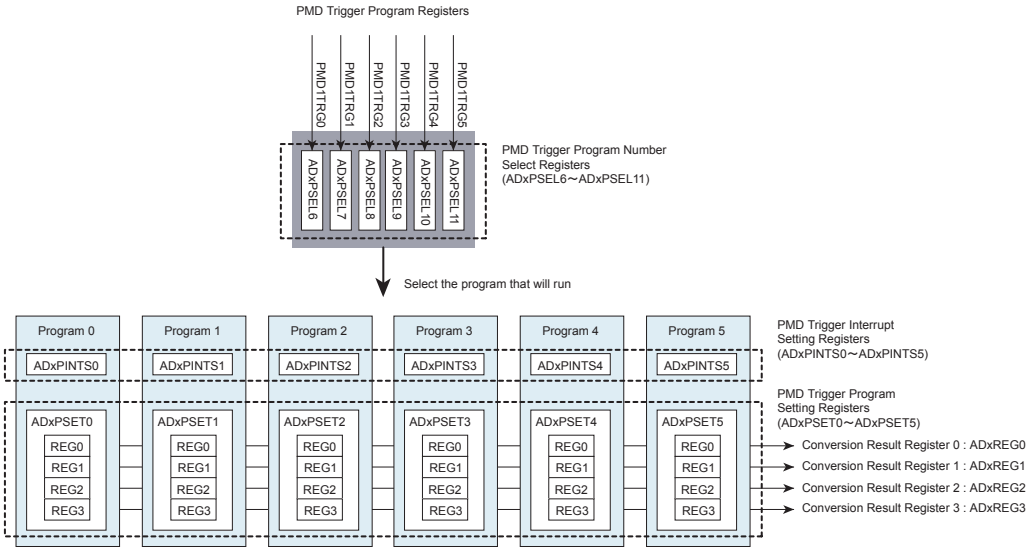


Figure 13-2 PMD Trigger Program Registers

13.4.22.1 ADxPSEL6 to ADxPSEL11(PMD Trigger Program Number Select Register 6 to 11)

ADxPSEL6:PMD Trigger Program Number Select 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	PENS6	-	-	-	-	PMDS6		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS6	R/W	PMD1TRG0 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS6[2:0]	R/W	Program number select (Refer to Table 13-1)

ADxPSEL7:PMD Trigger Program Number Select Register 7

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS7	-	-	-	-	PMDS7		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS7	R/W	PMD1TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS7[2:0]	R/W	Program number select (Refer to Table 13-1)

ADxPSEL8:PMD Trigger Program Number Select Register 8

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS8	-	-	-	-	PMDS8		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS8	R/W	PMD1TRG2 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS8[2:0]	R/W	Program number select (Refer to Table 13-1)

ADxPSEL9:PMD Trigger Program Number Select Register 9

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS9	-	-	-	-	PMDS9		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS9	R/W	PMD1TRG3 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS9[2:0]	R/W	Program number select (Refer to Table 13-1)

ADxPSEL10:PMD Trigger Program Number Select Register 10

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS10	-	-	-	-	PMDS10		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS10	R/W	PMD1TRG4 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS10[2:0]	R/W	Program number select (Refer to Table 13-1)

ADxPSEL11:PMD Trigger Program Number Select Register 11

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS11	-	-	-	-	PMDS11		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS11	R/W	PMD1TRG5 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS11[2:0]	R/W	Program number select (Refer to Table 13-1)

Table 13-1 Program number select

<PMDS6[2:0]>~ <PMDS11[2:0]>	
000	Program0
001	Program1
010	Program2
011	Program3
100	Program4
101	Program5
110	reserved
111	reserved

13.4.22.2 ADxPINTS0 to 5(PMD Trigger Interrupt Select Register 0 to 5)

ADxPINTS0:PMD Trigger Interrupt Select 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	-	-	-	-	-	-	INTSEL0	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL0[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 0.

ADxPINTS1:PMD Trigger Interrupt Select 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	-	-	-	-	-	-	INTSEL1	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL1[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 1.

ADxPINTS2:PMD Trigger Interrupt Select 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	-	-	-	-	-	-	INTSEL2	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL2[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 2.

ADxPINTS3:PMD Trigger Interrupt Select 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	-	-	-	-	-	-	INTSEL3	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL3[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 3.

ADxPINTS4:PMD Trigger Interrupt Select 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	-	-	-	-	-	-	INTSEL4	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL4[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 4.

ADxPINTS5:PMD Trigger Interrupt Select 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	-	-	-	-	-	-	INTSEL5	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL5[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 5.

13.4.22.3 ADxPSET0 to 5(PMD Trigger Program Register 0 to 5)

Each ADxPSETn (n=0 to 5:Program number) is composed of four sets that assume <AINSPnm [4:0]>, and <ENSPnm> in a couple. (m=0 to 3)(x=B : ADC Unit)

Setting the <ENSPnm> to "1" enables the <AINSPnm[4:0]> bits are used to select the AIN pin to be used. With these conditions, the AD conversion is started and then stored into a conversion result register.

ADxREGm ADxPSETn	m=0	m=1	m=2	m=3
n=0	<ENSP00> <AINSP00>	<ENSP01> <AINSP01>	<ENSP02> <AINSP02>	<ENSP03> <AINSP03>
n=1	<ENSP10> <AINSP10>	<ENSP11> <AINSP11>	<ENSP12> <AINSP12>	<ENSP13> <AINSP13>
n=2	<ENSP20> <AINSP20>	<ENSP21> <AINSP21>	<ENSP22> <AINSP22>	<ENSP23> <AINSP23>
n=3	<ENSP30> <AINSP30>	<ENSP31> <AINSP31>	<ENSP32> <AINSP32>	<ENSP33> <AINSP33>
n=4	<ENSP40> <AINSP40>	<ENSP41> <AINSP41>	<ENSP42> <AINSP42>	<ENSP43> <AINSP43>
n=5	<ENSP50> <AINSP50>	<ENSP51> <AINSP51>	<ENSP52> <AINSP52>	<ENSP53> <AINSP53>

Table 13-2 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:Reserved
0_0011	:Reserved
0_0100	:Reserved
0_0101	:Reserved
0_0110	:Reserved
0_0111	:Reserved
0_1000	:Reserved
0_1001	:AINB9
0_1010	:AINB10
0_1011	:AINB11
0_1100	:AINB12
0_1101	:Reserved
0_1110	:Reserved
0_1111	:Reserved
1_0000	:Reserved
1_0001 to 1_1111	:Reserved

ADxPSET0:PMD Trigger Program Register 0

	31	30	29	28	27	26	25	24
bit symbol	ENSP03	-	-	AINSP03				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP02	-	-	AINSP02				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP01	-	-	AINSP01				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP00	-	-	AINSP00				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP03	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write "00".
28-24	AINSP03[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
23	ENSP02	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write "00".
20-16	AINSP02[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
15	ENSP01	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write "00".
12-8	AINSP01[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
7	ENSP00	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write "00".
4-0	AINSP00[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".

ADxPSET1:PMD Trigger Program Register 1

	31	30	29	28	27	26	25	24
bit symbol	ENSP13	-	-	AINSP13				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP12	-	-	AINSP12				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP11	-	-	AINSP11				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP10	-	-	AINSP10				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP13	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write "00".
28-24	AINSP13[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
23	ENSP12	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write "00".
20-16	AINSP12[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
15	ENSP11	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write "00".
12-8	AINSP11[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
7	ENSP10	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write "00".
4-0	AINSP10[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".

ADxPSET2:PMD Trigger Program Register 2

	31	30	29	28	27	26	25	24
bit symbol	ENSP23	-	-	AINSP23				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP22	-	-	AINSP22				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP21	-	-	AINSP21				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP20	-	-	AINSP20				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP23	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write "00".
28-24	AINSP23[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
23	ENSP22	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write "00".
20-16	AINSP22[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
15	ENSP21	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write "00".
12-8	AINSP21[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
7	ENSP20	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write "00".
4-0	AINSP20[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".

ADxPSET3:PMD Trigger Program Register 3

	31	30	29	28	27	26	25	24
bit symbol	ENSP33	-	-	AINSP33				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP32	-	-	AINSP32				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP31	-	-	AINSP31				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP30	-	-	AINSP30				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP33	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write "00".
28-24	AINSP33[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
23	ENSP32	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write "00".
20-16	AINSP32[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
15	ENSP31	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write "00".
12-8	AINSP31[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
7	ENSP30	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write "00".
4-0	AINSP30[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".

ADxPSET4:PMD Trigger Program Register 4

	31	30	29	28	27	26	25	24
bit symbol	ENSP43	-	-	AINSP43				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP42	-	-	AINSP42				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP41	-	-	AINSP41				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP40	-	-	AINSP40				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP43	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write "00".
28-24	AINSP43[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
23	ENSP42	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write "00".
20-16	AINSP42[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
15	ENSP41	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write "00".
12-8	AINSP41[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
7	ENSP40	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write "00".
4-0	AINSP40[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".

ADxPSET5:PMD Trigger Program Register 5

	31	30	29	28	27	26	25	24
bit symbol	ENSP53	-	-	AINSP53				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP52	-	-	AINSP52				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP51	-	-	AINSP51				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP50	-	-	AINSP50				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP53	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write "00".
28-24	AINSP53[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
23	ENSP52	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write "00".
20-16	AINSP52[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
15	ENSP51	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write "00".
12-8	AINSP51[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".
7	ENSP50	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write "00".
4-0	AINSP50[4:0]	R/W	AIN select Refer to "Table 13-2 Select the AIN pin".

13.4.23 ADxTSET03 / ADxTSET47 / ADxTSET811 (Timer Trigger Program Registers)

AD conversion can be started by INTTB51 generated from Timer5(TMRB5) as a trigger. There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to "1" enables the ADxTSETm register. The <AINSTm[4:0]> are used to select the AIN pin to be used. The numbers of the Timer Trigger Program Registers correspond to those of the AD Conversion Result Registers. When finished this AD conversion, interrupt : INTADxTMR is generated.

(m=0 to 11),(x=B : ADC Unit)

Table 13-3 Select the AIN pin

<AINST0 [4:0]> to <AINST11 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:Reserved
0_0011	:Reserved
0_0100	:Reserved
0_0101	:Reserved
0_0110	:Reserved
0_0111	:Reserved
0_1000	:Reserved
0_1001	:AINB9
0_1010	:AINB10
0_1011	:AINB11
0_1100	:AINB12
0_1101	:Reserved
0_1110	:Reserved
0_1111	:Reserved
1_0000	:Reserved
1_0001 to 1_1111	:Reserved

ADxTSET03: Timer Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST3[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
23	ENST2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST2[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
15	ENST1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST1[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
7	ENST0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST0[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".

ADxTSET47: Timer Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST6	-	-	AINST6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST5	-	-	AINST5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST4	-	-	AINST4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST7[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
23	ENST6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST6[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
15	ENST5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST5[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
7	ENST4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST4[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".

ADxTSET811: Timer Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST10	-	-	AINST10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST9	-	-	AINST9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST8	-	-	AINST8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST11[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
23	ENST10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST10[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
15	ENST9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST9[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".
7	ENST8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST8[4:0]	R/W	AIN select Refer to "Table 13-3 Select the AIN pin".

13.4.24 ADxSSET03 / ADxSSET47 / ADxSSET811(Software Trigger Program Registers)

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. Setting the <ENSSm> to "1" enables the ADxSSETm register. The <AINSSm[4:0]> are used to select the AIN pin to be used. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers. When finished this AD conversion, interrupt :INTADxSFT is generated.

(m=0 to 11),(x=B : ADC Unit)

Table 13-4 Select the AIN pin

<AINSS0 [4:0]> to <AINSS11 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:Reserved
0_0011	:Reserved
0_0100	:Reserved
0_0101	:Reserved
0_0110	:Reserved
0_0111	:Reserved
0_1000	:Reserved
0_1001	:AINB9
0_1010	:AINB10
0_1011	:AINB11
0_1100	:AINB12
0_1101	:Reserved
0_1110	:Reserved
0_1111	:Reserved
1_0000	:Reserved
1_0001 to 1_1111	:Reserved

ADxSSET03: Software Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS2	-	-	AINSS2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS1	-	-	AINSS1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS0	-	-	AINSS0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS3[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
23	ENSS2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS2[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
15	ENSS1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS1[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
7	ENSS0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS0[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".

ADxSSET47: Software Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS6	-	-	AINSS6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS5	-	-	AINSS5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS4	-	-	AINSS4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS7[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
23	ENSS6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS6[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
15	ENSS5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS5[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
7	ENSS4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS4[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".

ADxSSET811: Software Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS10	-	-	AINSS10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS9	-	-	AINSS9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS8	-	-	AINSS8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS11[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
23	ENSS10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS10[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
15	ENSS9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS9[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".
7	ENSS8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS8[4:0]	R/W	AIN select Refer to "Table 13-4 Select the AIN pin".

13.4.25 ADxASET03 / ADxASET47 / ADxASET811(Constant Conversion Program Registers)

The ADCs allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to "1" enables the ADxASETm register. The <AINSAm[4:0]> are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m=0 to 11),(x=B : ADC Unit)

Table 13-5 Select the AIN pin

<AINSA0 [4:0]> to <AINSA11 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:Reserved
0_0011	:Reserved
0_0100	:Reserved
0_0101	:Reserved
0_0110	:Reserved
0_0111	:Reserved
0_1000	:Reserved
0_1001	:AINB9
0_1010	:AINB10
0_1011	:AINB11
0_1100	:AINB12
0_1101	:Reserved
0_1110	:Reserved
0_1111	:Reserved
1_0000	:Reserved
1_0001 to 1_1111	:Reserved

ADxASET03: Constant Conversion Program Registers03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA2	-	-	AINSA2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA1	-	-	AINSA1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA0	-	-	AINSA0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA3[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
23	ENSA2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA2[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
15	ENSA1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA1[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
7	ENSA0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA0[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".

ADxASET47: Constant Conversion Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA6	-	-	AINSA6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA5	-	-	AINSA5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA4	-	-	AINSA4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA7[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
23	ENSA6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA6[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
15	ENSA5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA5[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
7	ENSA4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA4[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".

ADxASET811: Cnstant Conversion Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSA11	-	-	AINSA11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA10	-	-	AINSA10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA9	-	-	AINSA9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA8	-	-	AINSA8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA11[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
23	ENSA10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA10[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
15	ENSA9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA9[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".
7	ENSA8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA8[4:0]	R/W	AIN select Refer to "Table 13-5 Select the AIN pin".

13.5 Operation Descriptions

13.5.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the VREFHB and VREFLB pins are used in ADC B.

Note 1: During AD conversion, do not change the output data of port J/K, to avoid the influence on the conversion result.

Note 2: AD conversion results might be unstable by the following conditions.

Input operation is executed.

Output operation is executed.

Output current of port varies.

Take a countermeasure such as averaging the multiple conversion results, to get precise value.

13.5.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD trigger (See "13.4.22 PMD Trigger Program Registers")
- Timer trigger (TMRB5) (See "13.4.23 Timer Trigger Program Registers.")
- Software trigger (See "13.4.24 Software Trigger Program Registers.")

These start triggers are given priorities as shown below.

PMD trigger 0 > ··· > PMD trigger 5 > Timer trigger > Software trigger > constant trigger

When the higher-priority trigger occurs while an AD conversion is in progress, the higher-priority trigger is handled stop the ongoing program and start AD conversion correspond to higher-priority trigger number.

When the PMD trigger occurs while an PMD triggered AD conversion is in progress, the PMD trigger is handled after the ongoing program is completed.

It has some delay from generation of trigger to start of AD conversion. The delay depends on the trigger. The following timing chart and table show the delay.

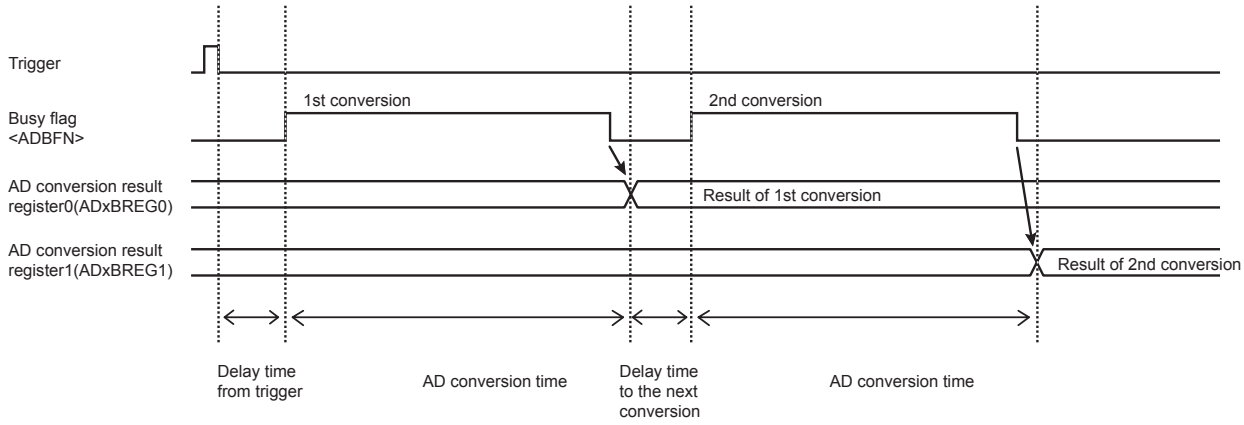


Figure 13-3 Timing chart of AD conversion

Table 13-6 AD conversion time and delay time (SCLK=40MHz, unit: μs)

		fsys = 40MHz	
	Trigger	MIN	MAX
Delay time from trigger [μs] (Note 1)	PMD	0.225	0.3
	TMRB	0.225	0.5
	Software, Constant	0.25	0.525
AD conversion time [μs]	–	1.85	
Delay time to the next conversion[μs] (Note2)	PMD	0.175	0.225
	TMRB, Software, Constant	0.175	0.425

Note 1: Delay time from trigger to start of AD conversion.

Note 2: Delay time to the 2nd or after conversion in plural conversions with one trigger.

13.5.3 AD Conversion Monitoring Function

The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function, set ADxCMPCR0<CMP0EN> or ADxCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by ADxCMPCR0<ADBIG0>/ADxCMCPR1<ADBIG1>, the interrupt (INTADxCPA for ADxCMPCR0, INTADxCPB for ADxCMPCR1) is generated. The comparison is executed at the timing of storing the conversion result into the register.

Note 1: The AD conversion result store flag <ADR0RF> to <ADR11RF> are not cleared by the comparison function.

Note 2: The comparison function differs from reading the conversion result by software. Therefore, if the next conversion is completed without reading the previous result, the overrun flag <OVR0> to <OVR11> are set.

13.6 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

13.6.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADxSSET03, ADxSSET47 and ADxSSET811.(Figure 13-4)

If the ADxMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.(Figure 13-5)

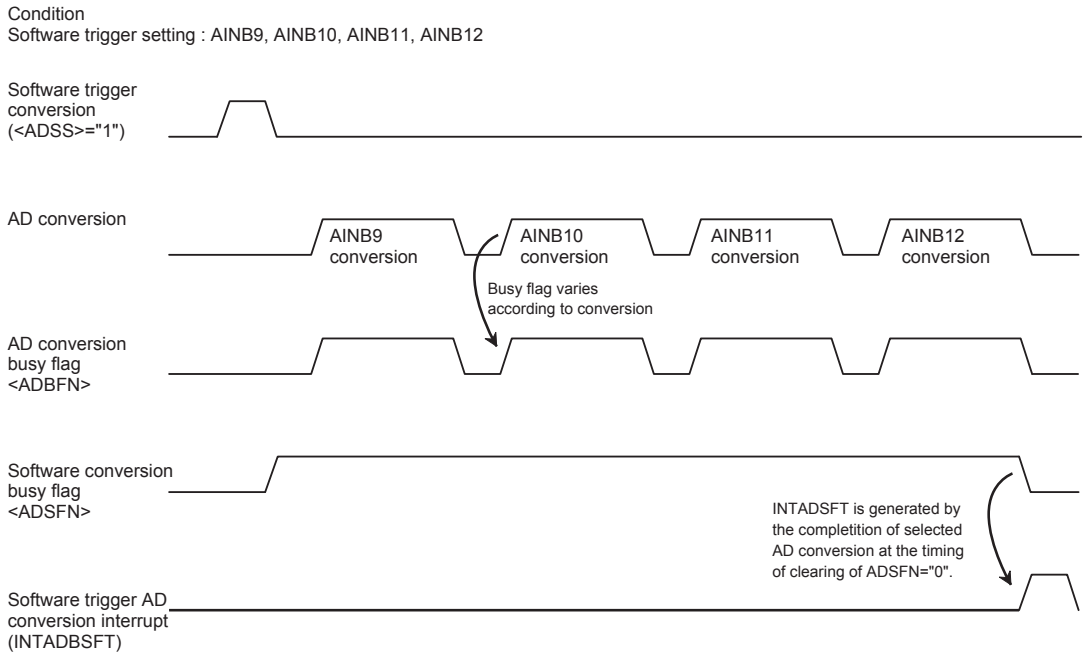


Figure 13-4 Software trigger AD conversion

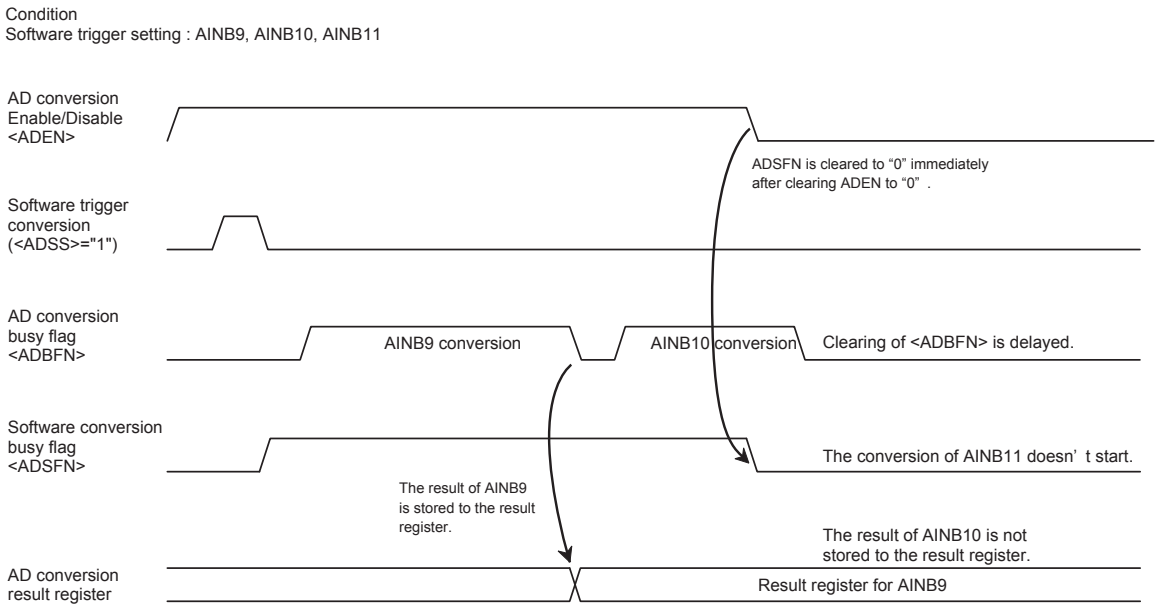


Figure 13-5 Writing "0" to <ADEN> during the software trigger AD conversion

13.6.2 Constant Conversion

In the constant conversion, if the next conversion completes without reading the previous result from the conversion result register, the overrun flag is set to "1". In this case, the previous conversion result in the conversion result register is overwritten by the next result. The overrun flag is cleared by reading of the conversion result.(Figure 13-6)

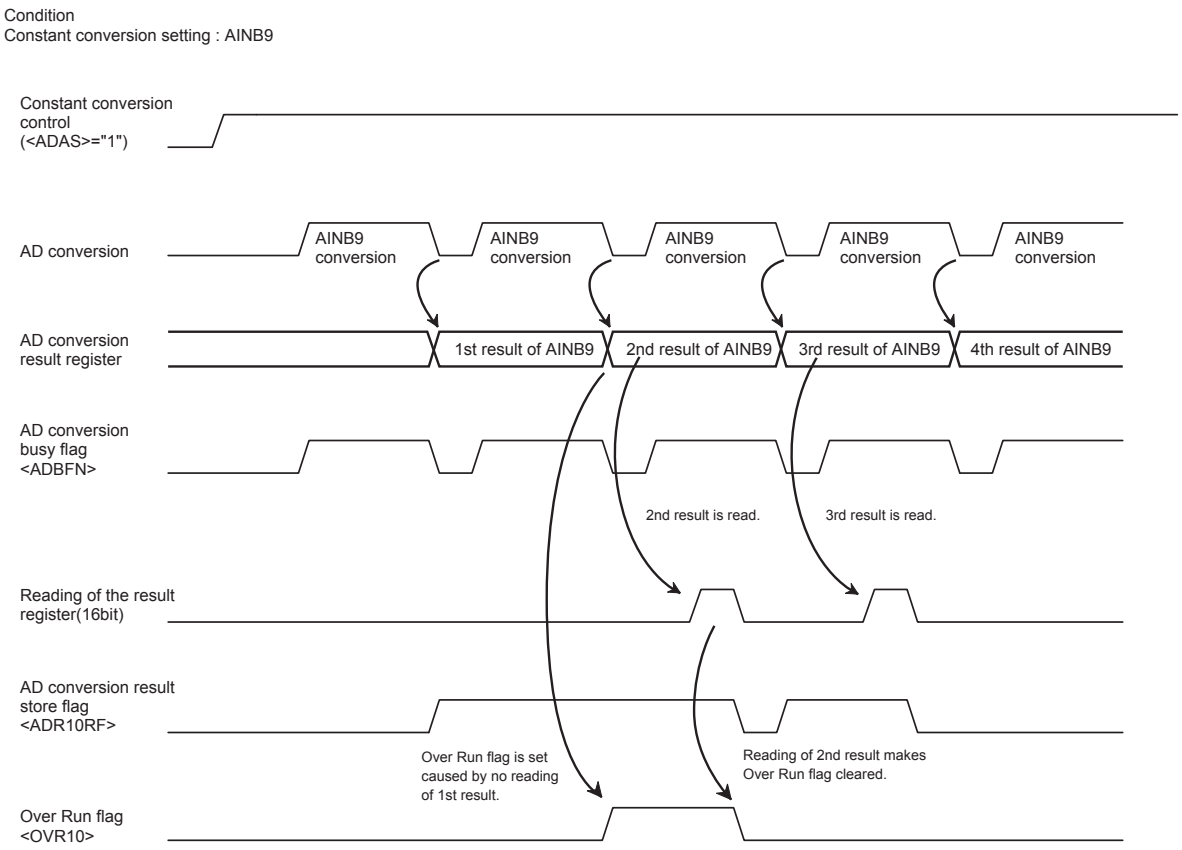


Figure 13-6 Constant conversion

13.6.3 AD conversion by trigger

When the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately and start AD conversion correspond to PMD trigger.(Figure 13-7) After the completion of conversion by PMD trigger, the software trigger conversion starts from the beginning programmed setting. When the timer trigger is occurred, also same response.(Figure 13-8)

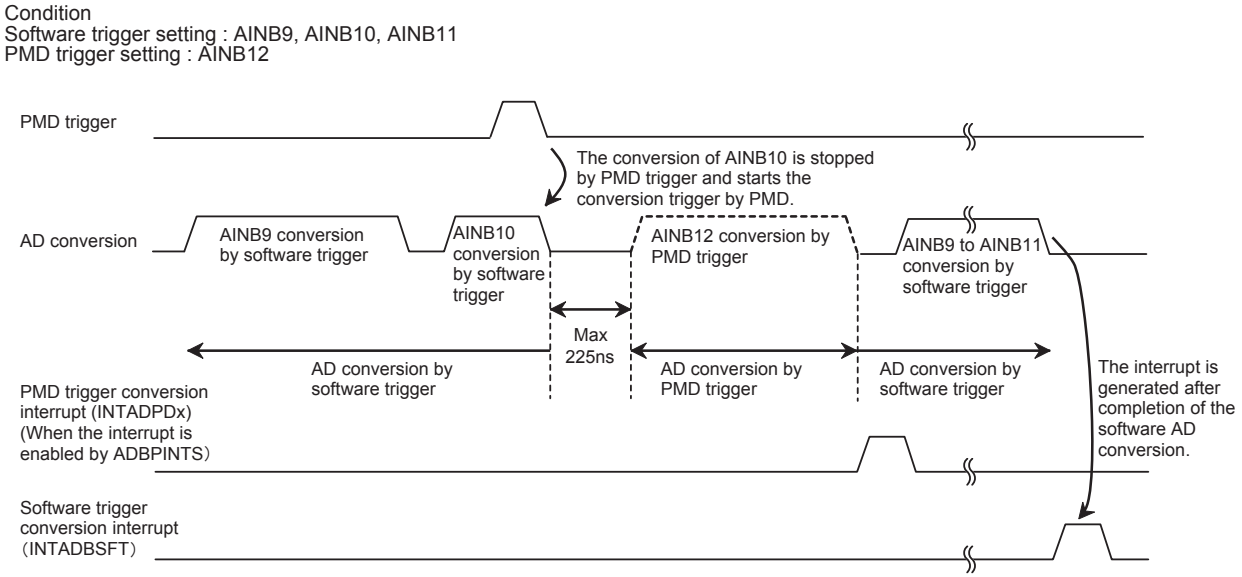


Figure 13-7 AD conversion by PMD trigger

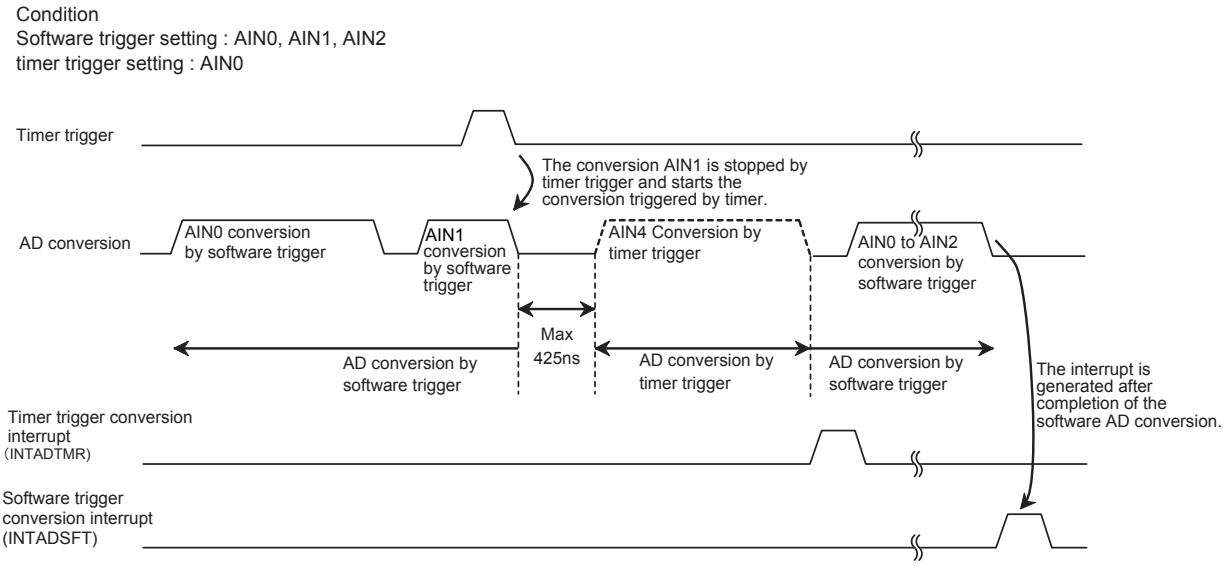


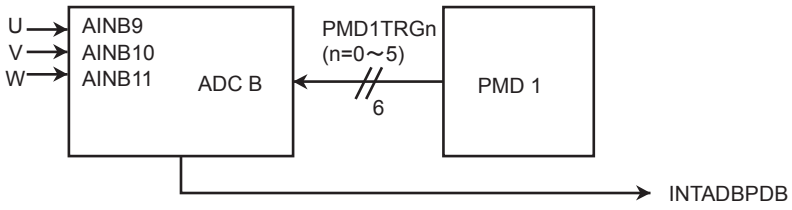
Figure 13-8 AD conversion by timer trigger

Note: When Timer trigger is not use, do not use INTTB51. Set TB5IM<TBIM1> to "1".

13.7 Usage Examples

13.7.1 Successive Conversion Using One PMD1(Three Shunts) and One ADC

The following shows a circuit diagram for AD conversion using one PMD1 for three shunts and one ADC.



Example ADC settings are shown below.

ADC UnitB

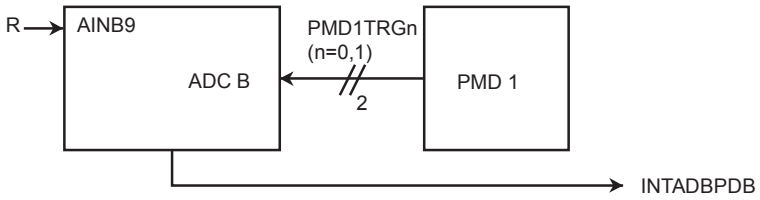
Program	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	B	B	B	B	B	B

Programs 0 to 5 are assigned to trigger inputs PMD1TRG0 to 5. "reg0" and "reg1" indicate the PMD Trigger Program Registers ADBPSETn[7:0] and ADBPSETn[15:8]. "U", "V" and "W" indicate the phases of a motor. AIN inputs are selected to obtain these phases.

When a trigger input occurs, AD conversion is performed based on reg0 and reg1 sequentially, and then the interrupt signal (INTADBPDB) is generated.

13.7.2 Successive Conversion Using One PMD1 (One Shunt) and One ADC

The following shows a circuit diagram for AD conversion using PMD1 for one shunt and one ADC.



Example ADC settings are shown below.

ADC UnitB

Trigger	PMD1	PMD1
	0	1
Program	0	1
reg0	R	-
reg1	-	R
INT	-	B

Programs 0 and 1 are assigned to two trigger signals from PMD1.

"reg0" and "reg1" indicate the PMD Trigger Program Registers ADBPSETn[7:0] and ADBPSETn[15:8]. "R" indicates a resistor, where the AIN input that is connected to that resistor is set.

When a trigger input occurs, the ADC is started to execute programs 0 and 1 sequentially. When program 1 is completed, the interrupt (INTADBPDB) is generated.

14. Motor Control Circuit (PMD: Programmable Motor Driver)

The TMPM3U0FSDMG contains 1 channel programmable motor driver (PMD). The PMD of this product can control a three-phase motor and an analog/digital converter (ADC). Pulse-width modulation circuits, conduction control. The synchronous trigger generation circuit can command the AD con-verter to start ADC conversion.

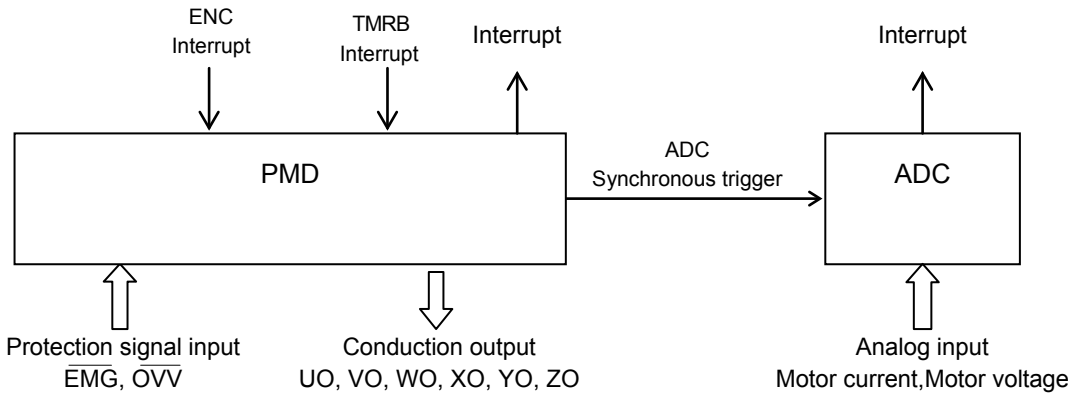


Figure 14-1 Block Diagram of Functions related to Motor Control

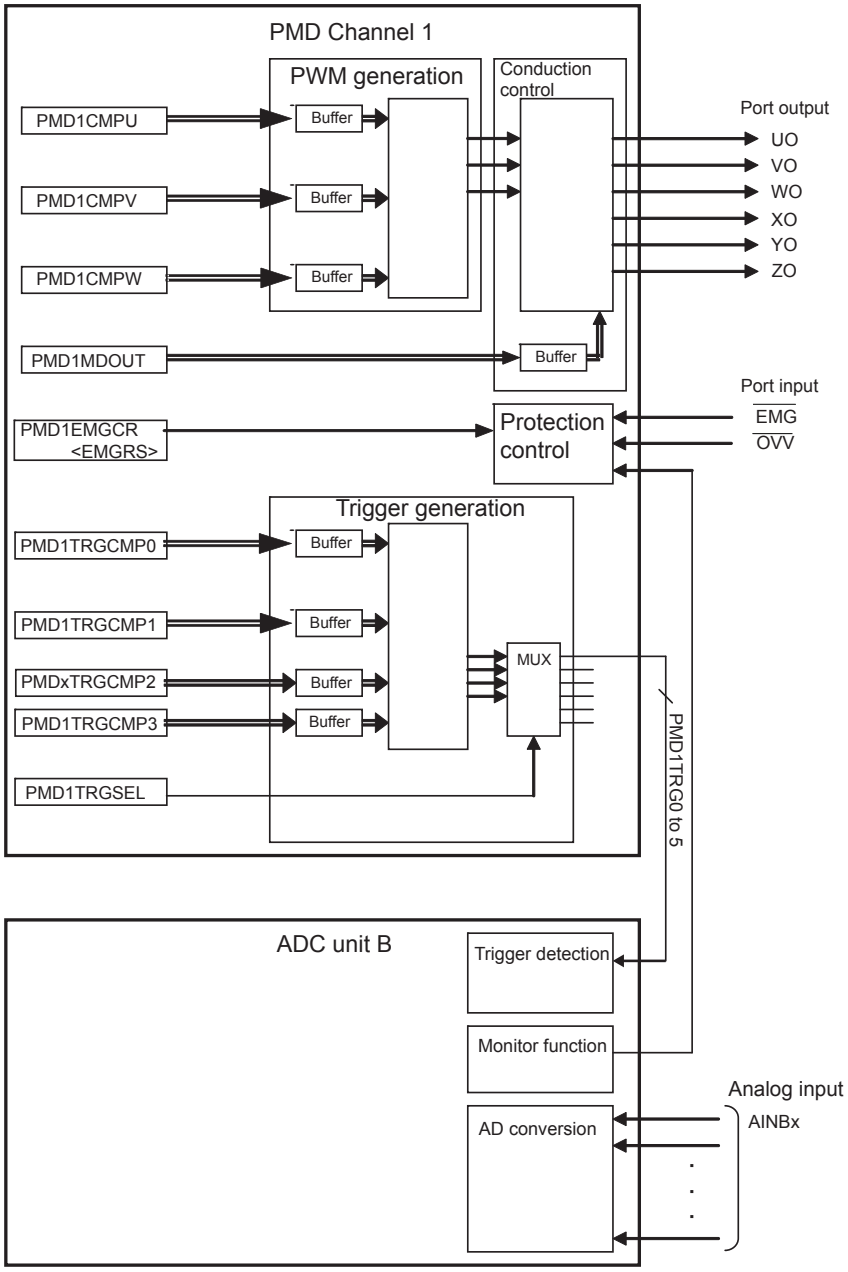


Figure 14-2 Related diagram of Motor control circuit and A/D converter

14.1 PMD Input/Output Signals

The table below shows the signals that are input to and output from PMD.

Table 14-1 Input/Output Signals

Channe	Pin Name	**PMD Signal Name**	**Description**
PMD1	PG6/$\overline{\text{EMG}}$/$\overline{\text{OVV}}$	$\overline{\text{EMG}}$/$\overline{\text{OVV}}$	EMG state signal /OVV state signal
	PG5/UO	UO	U-phase output
	PG4/XO	XO	X-phase output
	PG3/VO	VO	V-phase output
	PG2/YO	YO	Y-phase output
	PG1/WO	WO	W-phase output
	PG0/ZO	ZO	Z-phase output

14.2 PMD Circuit configuration

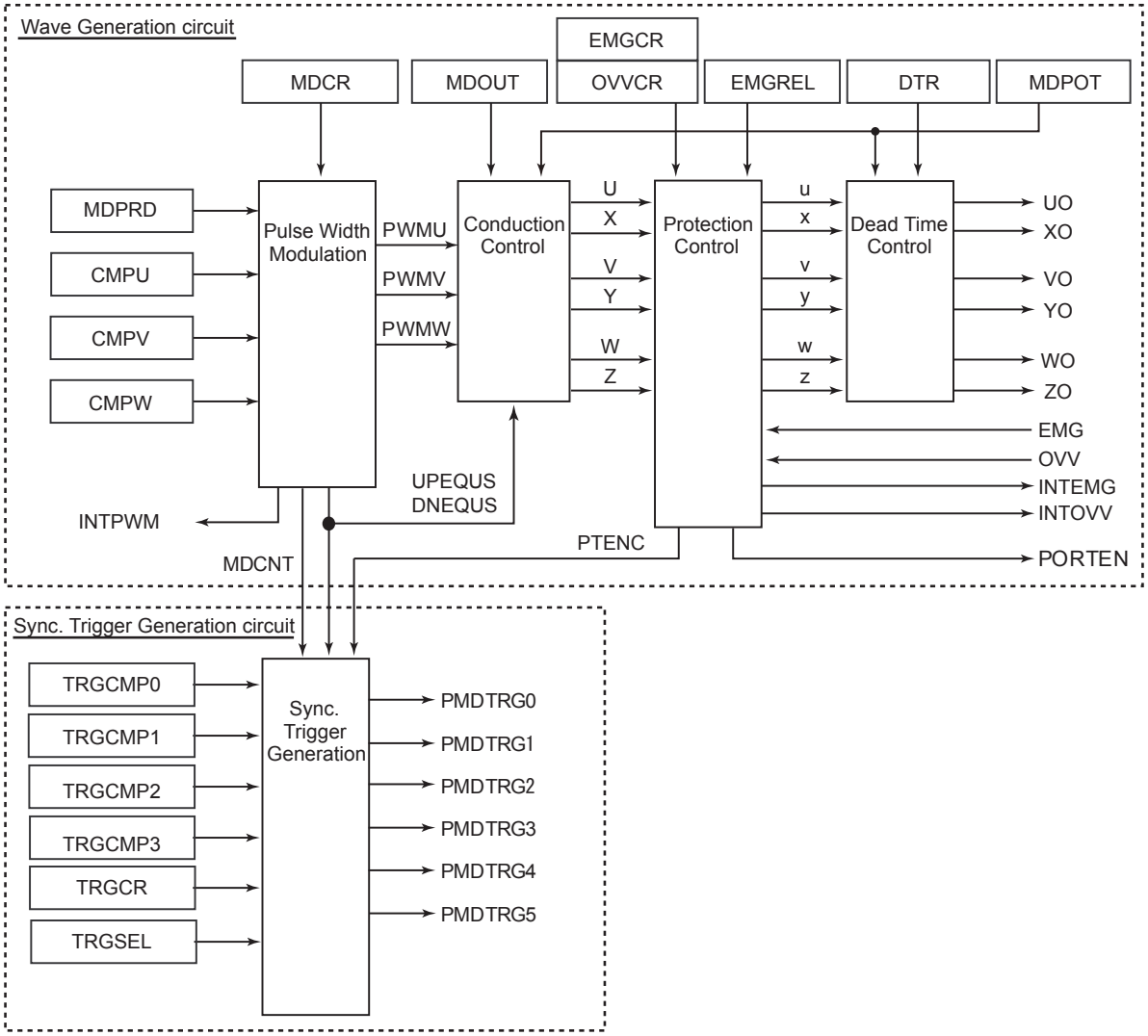


Figure 14-3 Block diagram of PMD Circuit

The PMD circuit consists of two blocks of a wave generation circuit and a sync trigger generation circuit. The wave generation circuit includes a pulse width modulation circuit, a conduction control circuit, a protection control circuit, a dead time control circuit.

- The pulse width modulation circuit has the common PWM carrier waveform and generates independent 3-phase PWM waveforms.
- The conduction control circuit determines the output pattern for each of the upper and lower sides of the U, V and W phases.
- The protection control circuit controls emergency output stop by EMG input and OVV input.
- The dead time control circuit prevents a short circuit which may occur when the upper side and lower side are switched.
- The sync trigger generation circuit generates sync trigger signals to the AD converter.

14.3 PMD Registers

The table below shows the registers related to the PMD.

Base Address = 0x4005_0480

Register Name		Address(Base+)
PMD Enable Register	PMD1MDEN	0x0000
Port Output Mode Register	PMD1PORTMD	0x0004
PMD Control Register	PMD1MDCR	0x0008
PWM Counter Status Register	PMD1CNTSTA	0x000C
PWM Counter Register	PMD1MDCNT	0x0010
PWM Period Register	PMD1MDPRD	0x0014
PMD Compare U Register	PMD1CMPU	0x0018
PMD Compare V Register	PMD1CMPV	0x001C
PMD Compare W Register	PMD1CMPW	0x0020
Reserved	-	0x0024
PMD Conduction Control Register	PMD1MDOUT	0x0028
PMD Output Setting Register	PMD1MDPOT	0x002C
EMG Release Register	PMD1EMGREL	0x0030
EMG Control Register	PMD1EMGCR	0x0034
EMG Status Register	PMD1EMGSTA	0x0038
OVV Control Register	PMD1OVVCR	0x003C
OVV Status Register	PMD1OVVSTA	0x0040
Dead Time Register	PMD1DTR	0x0044
Trigger Compare 0 Register	PMD1TRGCMP0	0x0048
Trigger Compare 1 Register	PMD1TRGCMP1	0x004C
Trigger Compare 2 Register	PMD1TRGCMP2	0x0050
Trigger Compare 3 Register	PMD1TRGCMP3	0x0054
Trigger Control Register	PMD1TRGCR	0x0058
Trigger Output Mode Setting Register	PMD1TRGMD	0x005C
Trigger Output Select Register	PMD1TRGSEL	0x0060
Trigger Update Timing Setting Register	PMD1TRGSYNCR	0x0064
Reserved	-	0x0068
Reserved	-	0x006C
Reserved	-	0x0070
Reserved	-	0x0074
Reserved	-	0x0078
Reserved	-	0x007C

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

14.3.1 PMD1MDEN(PMD Enable Register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	PWMEN	R/W	Enables or disables waveform synthesis. 0: Disable 1: Enable Note: When the port is set to a function output (PWM output), the port disables output (high impedance) by setting <PWMEN> = "0". Note: Before enabling the PMD, Setting <PWMEN>="1"(enable) other relevant settings, such as output port polarity.

14.3.2 PMD1PORTMD(Port Output Mode Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1-0	PORTMD[1:0]	R/W	Port control setting when a tool break occurs 00: Upper phases = High-z / lower phases = High-z 01: Upper phases = High-z / lower phases = PMD output 10: Upper phases = PMD output / lower phases = High-z 11: Upper phases = PMD output / lower phases = PMD output Sets the port output for both upper phase (UO/VO/WO) and the lower phase (XO/YO/ZO) when a tool break occurs in the use of ports for function output (PWM output). When a tool break occurs while "High-Z" is selected, the ports are disabled to output (high impedance). In other cases, external port outputs depend on PMD outputs.

Note 1: When <PWMEN>=0, output ports are disabled to output (high impedance) regardless of the PORTMD setting.

Note 2: When an EMG input occurs, port outputs are controlled depending by setting the PMD1EMGCR<EMGMD[1:0]>.

14.3.3 Pulse Width Modulation Circuit

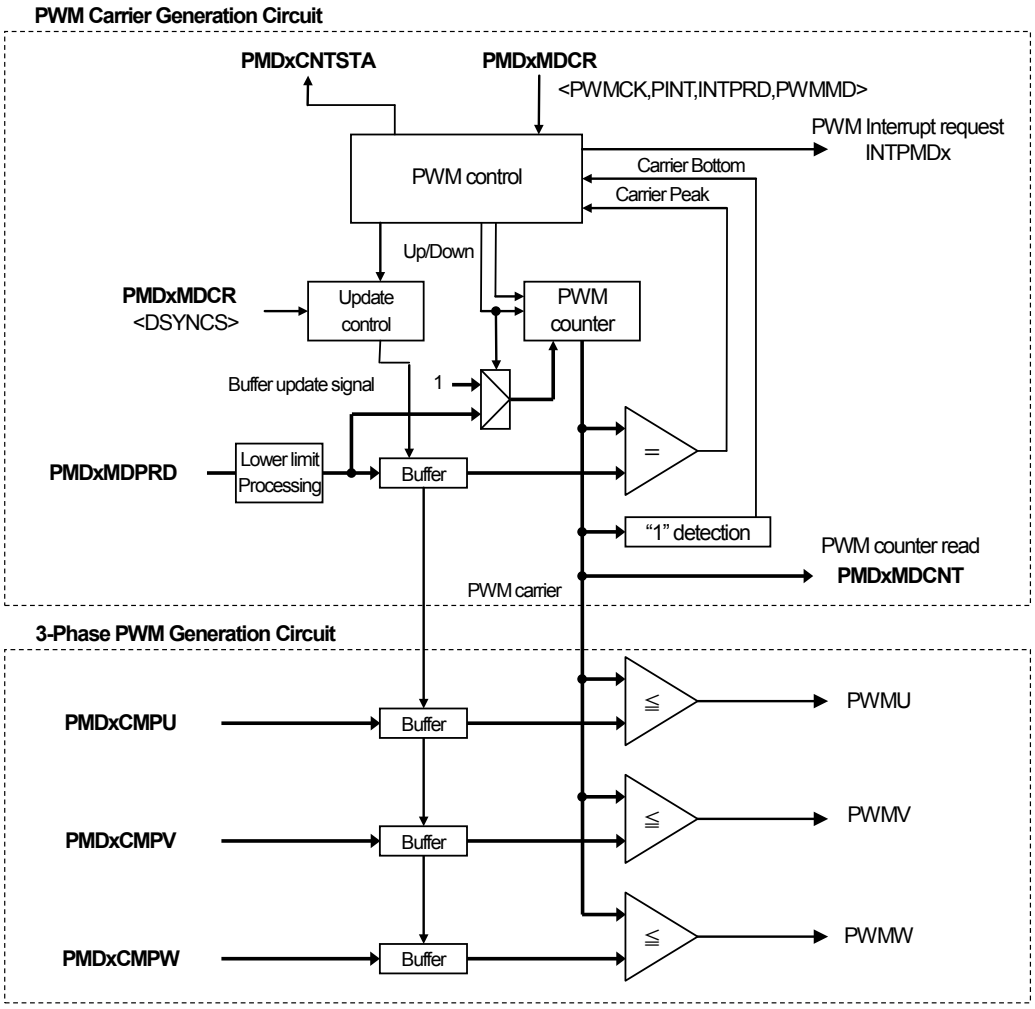


Figure 14-4 Pulse Width Modulation Circuit

The pulse width modulation circuit has a 16-bit PWM up-/down-counter and generates PWM carrier waveforms with a resolution of 25 nsec at 40 MHz. The PWM period extension mode (PMD1MDCR<PWMCK> ="1") is also available. When this mode is selected, the PWM counter generates PWM carrier waveforms with a resolution of 100 nsec at 40MHz.

The PWM carrier waveform mode can be selected from mode 0 (edge-aligned PWM, sawtooth wave modulation) and mode 1 (center-aligned PWM, triangular wave modulation). (Refer to "Figure 14-5 PWM Waveforms".) In the triangular wave mode, PWM waveform can be selected from the center PWM, the fixed falling edge PWM and the fixed rising edge PWM. (Refer to "Figure 14-6 Waveforms of PWM triangular wave carrier using fixed edge".)

1. Setting the PWM period

The PWM period is determined by the PMD1MDPRD register. This register is double-buffered. The subsequent stage buffer is updated at every PWM period. It is also possible to update at every half PWM period. (Refer to "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing".)

$$\text{Sawtooth wave PWM : PMD1MDPRD register Value} = \frac{\text{Oscillation frequency[Hz]}}{\text{PWM frequency[Hz]}}$$

$$\text{Triangular wave PWM : PMD1MDPRD register value} = \frac{\text{Oscillation frequency[Hz]}}{\text{PWM frequency[Hz] x 2}}$$

2. Compare function

The pulse width modulation circuit generates PWM waveforms of the desired duty by comparing the magnitude of the PWM compare registers (PMD1CMPU/V/W) and the PWM carrier which is generated by the PWM counter (PMD1MDCNT <MDCNT[15:0]>).

The PWM compare register of each phase has a double-buffered register. The PWM compare register value is loaded into the subsequent stage buffer at every PWM period. It is also possible to update at every half a PWM period. (Refer to "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing".)

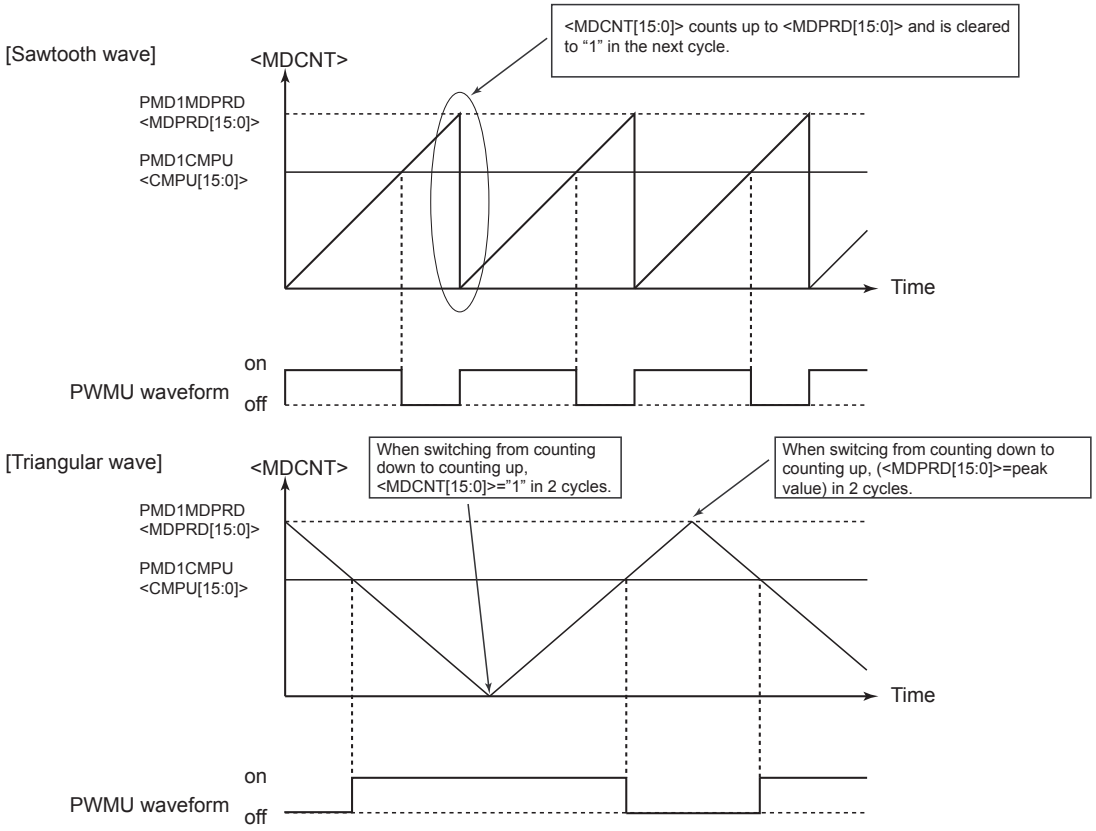


Figure 14-5 PWM Waveforms

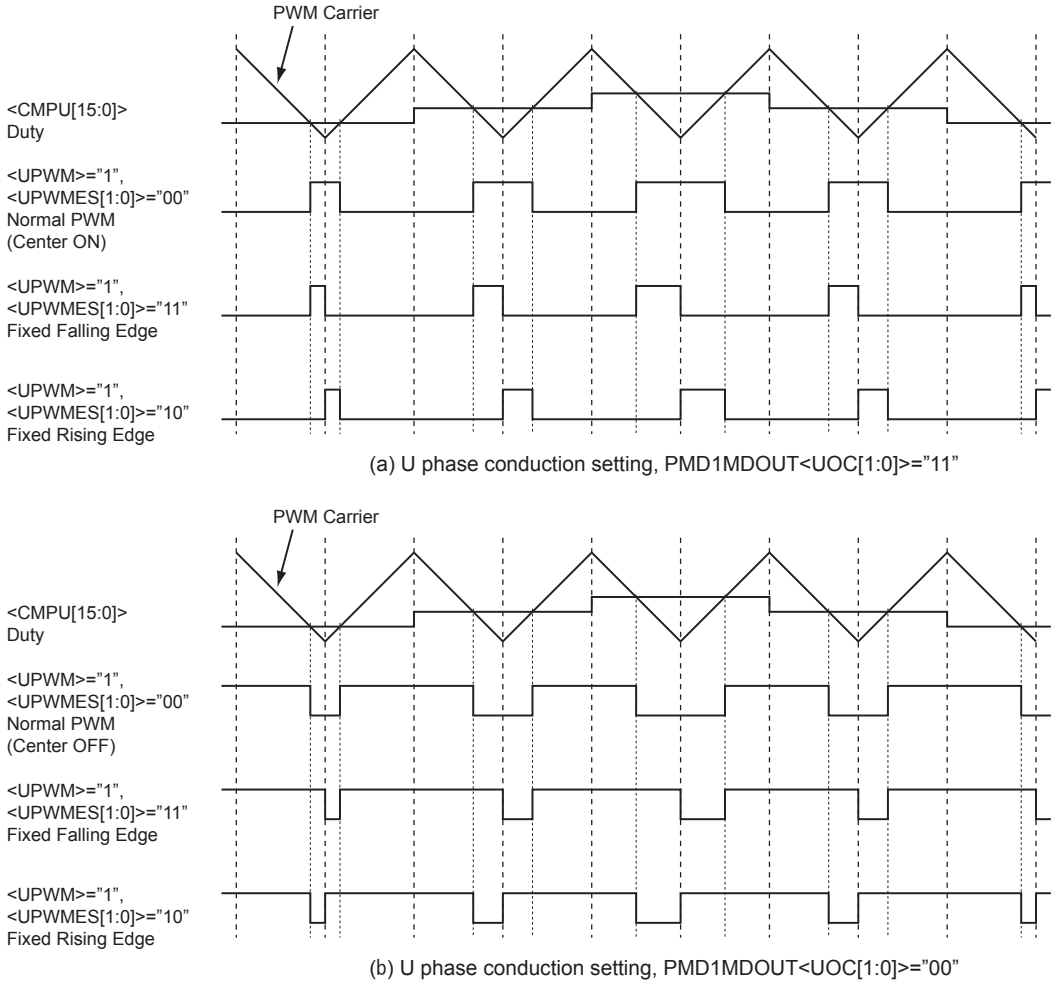


Figure 14-6 Waveforms of PWM triangular wave carrier using fixed edge

3. Waveform mode

Three-phase PWM waveforms can be generated in the following two modes:

1. 3-phase independent mode:

Each of the PWM compare registers for the three phases is set independently to generate independent PWM waveforms for each phase. This mode is used to generate drive waveforms such as sinusoidal waves.

2. 3-phase common mode:

Only the U-phase PWM compare register is set to generate identical PWM waveforms for all the three phases. This mode is used for rectangular wave drive of brushless DC motors.

4. Interrupt processing

The pulse width modulation circuit generates PWM interrupt requests in synchronization with PWM waveforms. Interrupt request timing can be selected either at PWM carrier peak or at PWM carrier bottom.

The PWM interrupt period can be set to half a PWM period, one PWM period, two PWM periods or four PWM periods.

14.3.3.1 PMD1MDCR (PMD Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	WPWMES		VPWMES		UPWMES		DSYNCS	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-14	WPWMES[1:0]	R/W	W-phase edge setting 00: Edge unfixed (center-aligned PWM) 01: Reseved 10: PWM rising-edge fixed (to the PWM carrier bottom) 11: PWM falling-edge fixed (to the PWM carrier bottom) Note: Valid when triangular carrier PWM is selected (<PWMMD> ="1").
13-12	VPWMES[1:0]	R/W	V-phase edge setting 00: Edge unfixed (center-aligned PWM) 01: Reseved 10: PWM rising-edge fixed (to the PWM carrier bottom) 11: PWM falling-edge fixed (to the PWM carrier bottom) Note: Valid when triangular carrier PWM is selected (<PWMMD> ="1").
11-10	UPWMES[1:0]	R/W	U-phase edge setting 00: Edge unfixed (center-aligned PWM) 01: Reseved 10: PWM rising-edge fixed (to the PWM carrier bottom) 11: PWM falling-edge fixed (to the PWM carrier bottom) Note: Valid when triangular carrier PWM is selected (<PWMMD> ="1").
9-8	DSYNCS[1:0]	R/W	Double buffer update timing for the duty compare register and PWM period register. 00: Depends on interrupt cycle setting (refer to the Table 14-2) Updates at the carrier peak and carrier bottom when 0.5 PWM period is selected (<INTPRD> ="00"). Otherwise, updates at the carrier peak. 01: Updates at PWM carrier bottom 10: Updates at PWM carrier peak 11: Updates at both PWM carrier peak and bottom Note1: Updates at carrier peak when sawtooth wave carrier is selected (<PWMMD>="0") regardless of the setting. Note2: When PMD1MDEN<PWMEN>="0", updates asynchronously regardless of setting.
7	-	R	Read as 0.
6	PWMCK	R/W	PWM period extension mode 0: Normal period 1: 4 × period Sets the counting cycle of the PWM counter. Normal cycle setting: sawtooth wave 25ns/triangular wave 50ns at fsys=40MHz Quadruple cycle setting: sawtooth wave 100ns / triangular wave 200ns at fsys=40MHz
5	SYNTMD	R/W	Port output mode Port outputs are controlled by a combination of <nOC>,<nPWM>,<POLH>,<POLL> and <SYNTMD> (refer to Table 14-4).

Bit	Bit Symbol	Type	Function
4	DTYMD	R/W	Duty mode 0: 3-phase common mode 1: 3-phase independent mode This bit selects whether to make duty setting independently for each phase or to use the PMD1CMPU register as 3-phase common.
3	PINT	R/W	PWM interrupt request timing 0: Interrupt request occurs at PWM carrier bottom (PMD1MDCNT<MDCNT[15:0]> = 0x0001). 1: Interrupt request occurs at PWM carrier peak (PMD1MDCNT<MDCNT[15:0]> = <MDPRD[15:0]>). Note1: Interrupt request occurs at carrier peak when the PWM carrier is sawtooth wave (<PWMMD>="0"). Note2: Interrupt request occurs both at carrier peak and carrier bottom when the interrupt cycle is 0.5 period (<INTPRD>="00").
2-1	INTPRD[1:0]	R/W	PWM interrupt request cycle 00: Interrupt request at every 0.5 PWM period Note1: PWM interrupt request cycle can be configured only when the PWM carrier is triangular wave (<PWMMD>="1") Note2: The double buffer of the compare registers (PMD1CMPU/V/W) and the cycle register (PMD1MDPRD) are updated by peak and bottom of the PWM carrier. 01: Interrupt request at every PWM period 10: Interrupt request at every two PWM periods 11: Interrupt request at every four PWM periods This field selects the PWM interrupt request period from 0.5 PWM period, one PWM period, two PWM periods and four PWM periods.
0	PWMMD	R/W	PWM carrier waveform 0: PWM mode 0 (edge-aligned PWM and sawtooth wave) 1: PWM mode 1(center-aligned PWM and triangular wave)

Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing

Setting		Update Timing
<DSYNCS[1:0]>	<INTPRD[1:0]>	
00	1x	Updates at PWM carrier peak
	x1	Updates at PWM carrier peak
	00	Updates at PWM carrier peak and PWM carrier bottom
01	xx	Updates at PWM carrier bottom
10	xx	Updates at PWM carrier peak
11	xx	Updates at PWM carrier peak and PWM carrier bottom

x : Don’t care

14.3.3.2 PMD1CNTSTA (PWM Counter Status Register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	UPDWN	R	PWM counter flag 0: Up-counting 1: Down-counting This bit indicates whether the PWM counter is up-counting or down-counting. Note: The PWM carrier is a sawtooth wave (PMD1MDCR<PWMMD>="0"), a zero is always read.

14.3.3.3 PMD1MDCNT(PWM Counter Register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	MDCNT[15:0]	R	PWM counter A value can be read from the up-down counter generating a PWM carrier wave. Counter resolution: 25ns at fsys = 40MHz Note1: When a quadruple cycle mode is selected (PMD1MDCR<PWMCK>="1"), time resolution of the counter is 100ns @ fsys=40MHz. Note2: Depending on the setting of the PWM carrier (PMD1MDCR<PWMMD>), the PWM counter values when PMD is disabled (PMD1MDEN<PWMEN>="0") are as follows: In case of PMD1MDCR<PWMMD>="0" : 0x0001 In case of PMD1MDCR<PWMMD>="1" : the value of PMD1MDPRD<MDPRD[15:0]>

14.3.3.4 PMD1MDPRD(PWM Period Register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	MDPRD[15:0]	R/W	PWM period <MDPRD[15:0]> ≥ 0x010 By the combination of the PWM period extension mode (PMD1MDCR<PWMCK>) and the PWM carrier waveform <PWMMD>, the PWM cycle can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <MDPRD> × 1/fsys <PWMMD>="1" : <MDPRD> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <MDPRD> × 4/fsys <PWMMD>="1" : <MDPRD> × 8/fsys Note: If <MDPRD[15:0]> is set to a value less than 0x0010, it is automatically assumed to be 0x0010. (The register retains the actual value that is written.)

Note 1: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 2: Since the PMD1MDPRD register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing".

Note 4: Read value is the first buffer value (the latest data set via a bus).

14.3.3.5 PMD1CMPU (PWM Compare Registers of U Phase)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	CMPU[15:0]	R/W	PWM pulse width of U Phase 0x0000 through 0xFFFF Note: When <CMPU> > <MDPRD>, the duty is 100%. <CMPU[15:0]> are compare registers for determining the output pulse width of the U phases. These registers are double-buffered. Pulse width is determined by comparing the subsequent stage buffer and the PWM counter to evaluate which is small or large. By the combination of the PWM period extension mode (PMD1MDCR<PWMCK>) and the PWM carrier waveform (<PWMMD>), the pulse width can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <CMPU> × 1/fsys <PWMMD>="1" : <CMPU> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <CMPU> × 4/fsys <PWMMD>="1" : <CMPU> × 8/fsys

Note 1: Read value is the first buffer value (the latest data set via a bus).
Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
Note 3: Since the PMD1CMPU register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing".

14.3.3.6 PMD1CMPV (PWM Compare Registers of V Phase)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	CMPV[15:0]	R/W	PWM pulse width of V Phase 0x0000 through 0xFFFF Note: When <CMPV> > <MDPRD>, the duty is 100%. <CMPV[15:0]> are compare registers for determining the output pulse width of the V phases. These registers are double-buffered. Pulse width is determined by comparing the subsequent stage buffer and the PWM counter to evaluate which is small or large. By the combination of the PWM period extension mode (PMD1MDCR<PWMCK>) and the PWM carrier waveform (<PWMMD>), the pulse width can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <CMPV> × 1/fsys <PWMMD>="1" : <CMPV> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <CMPV> × 4/fsys <PWMMD>="1" : <CMPV> × 8/fsys

Note 1: Read value is the first buffer value (the latest data set via a bus).
Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
Note 3: Since the PMD1CMPV register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing".

14.3.3.7 PMD1CMPW (PWM Compare Registers of W Phase)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	CMPW[15:0]	R/W	PWM pulse width of W Phase 0x0000 through 0xFFFF Note: When <CMPW> > <MDPRD>, the duty is 100%. <CMPW[15:0]> are compare registers for determining the output pulse width of the W phases. These registers are double-buffered. Pulse width is determined by comparing the subsequent stage buffer and the PWM counter to evaluate which is small or large. By the combination of the PWM period extension mode (PMD1MDCR<PWMCK>) and the PWM carrier waveform (<PWMMD>), the pulse width can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <CMPW> × 1/fsys <PWMMD>="1" : <CMPW> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <CMPW> × 4/fsys <PWMMD>="1" : <CMPW> × 8/fsys

Note 1: Read value is the first buffer value (the latest data set via a bus).
Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
Note 3: Since the PMD1CMPW register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W Buffer Update Timing".

14.3.4 Conduction Control Circuit

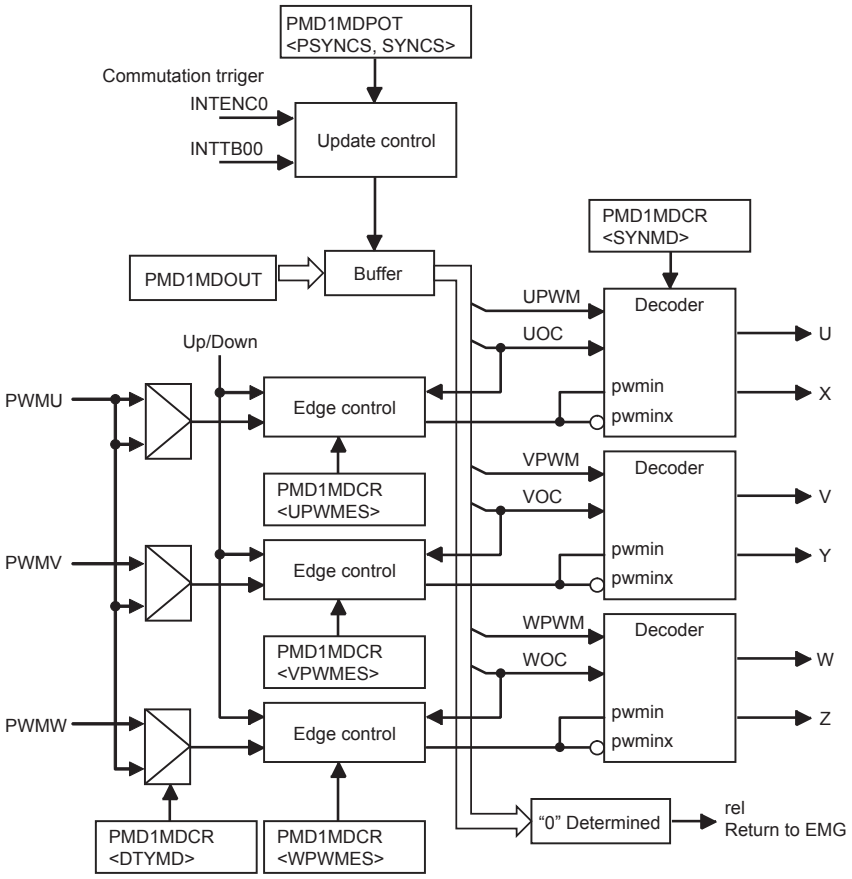


Figure 14-7 Conduction Control Circuit

The conduction control circuit performs the output port control according to the settings made in the output control register PMD1MDOUT and the output setting register PMD1MDPOT. PMD1MDOUT register is double-buffered and update timing can be select as synchronous or asynchronous to PWM. Update timing synchronizing with trigger input can also be selected. (For details of up-date timing, refer to "Table 14-3 Update Timing of the PMD1MDOUT buffer".)

Using PMD1MDPOT<POLH>,<POLL>, six output ports can be set to low-active or high-active on upper-phase output (UO,VO,WO) or lower-phase output (XO,YO,ZO) respectively. In addition, <WPWM>, <VPWM>, <UPWM> of the PMD1MDOUT register selects PWM or High/Low output for each of the U, V and W phases. When PWM output is selected, PWM waveforms are output. When High/Low output is selected, output is fixed to either a High or Low level. Each output is set to high or low by <WOC>, <VOC>, <UOC> of the register PMD1MDOUT.

"Table 14-4 Port Outputs according to the <UOC>,<VOC>,<WOC>,<UPWM>,<VPWM> and <WPWM> settings" shows port outputs setting according to port output setting in the PMD1MDOUT register and PMD1MDPOT register, and port output polarity setting in the port output mode of PMD1MDCR register.

Table 14-3 Update Timing of the PMD1MDOUT buffer

		PSYNCS setting			
		00	01	10	11
SYNCS setting	00	Constant update	PWM carrier bottom	PWM carrier peak	PWM carrier peak and PWM carrier bottom
	01	When INTENC is arisen.	The first PWM carrier bottom every time when INTENC is arisen.	The first PWM carrier peak every time when INTENC is arisen.	Either the first PWM carrier peak or the first carrier bottom every time when INTENC is arisen.
	10	When INTTB00 is arisen.	The first PWM carrier bottom every time when INTTB00 is arisen.	The first PWM carrier peak every time when INTTB00 is arisen.	Either the first PWM carrier peak or the first carrier bottom every time when INTTB00 is arisen.
	11	-	-	-	-

Note: IF PMD is disabled (PMD1MDCR<PMWEN>="0"), the retained trigger condition is cleared.

14.3.4.1 PMD1MDPOT (PMD Output Setting Register)

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

Bit	Bit Symbol	Type	Function
31-10	-	R	Read as 0.
9-8	SYNCS[1:0]	R/W	Selects PMD1MDOUT transfer timing (trigger synchronous setting). 00: asynchronous 01: when INTENC (ENC interrupt request) occurs 10: when INTTB00 (TMRB interrupt request) occurs 11: Reserved Selects the subsequent stage buffer update timing of the conduction control register. Note1: By the combination of the settings for <PSYNC> and <SYNCS>, the buffer update timing can be determined (refer to the "Table 14-3 Update Timing of the PMD1MDOUT buffer"). Note2: When PMD is disabled (PMD1MDEN<PWMEN>="0"), the timing is asynchronous regardless of settings.
7-4	-	R	Read as 0.
3	POLH	R/W	Selects the output polarity of the upper phase output (UO, VO, WO). 0: Active low 1: Active high
2	POLL	R/W	Selects the output polarity of the lower phase output (XO, YO, ZO). 0: Active low 1: Active high
1-0	PSYNCS[1:0]	R/W	Selects PMD1MDOUT transfer timing (PWM synchronous setting). 00: asynchronous to PWM The setting is applied to the port output at the same time that the PMD1MDOUT registers. 01: Carrier bottom (when <MDCNT[15:0]>="1") 10: Carrier peak (<MDCNT[15:0]>=<MDPRD[15:0]>) 11: Carrier peak and carrier bottom Selects the subsequent stage buffer update timing of the conduction control register. Note1: When the PWM carrier is sawtooth wave, the buffer update timing is the carrier peak, except <PSYNCS>="00" . Note2: By the combination of the settings for <PSYNC> and <SYNCS>, the buffer update timing can be determined (refer to the "Table 14-3 Update Timing of the PMD1MDOUT buffer"). Note3: When PMD is disabled (PMD1MDEN<PWMEN>="0"), the timing is asynchronous regardless of settings.

Note: This field must be set while PMD1MDEN<PWMEN>="0".

14.3.4.2 PMD1MDOUT(PMD Conduction Control Register)

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

Bit	Bit Symbol	Type	Function
31-11	-	R	Read as 0.
10	WPWM	R/W	W-phase PWM output setting 0: High/Low Output 1: PWM Output Port output is controlled by the combination of <WOC>,<WPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 14-4).
9	VPWM	R/W	V-phase PWM output setting 0: High/Low Output 1: PWM Output Port output is controlled by the combination of <VOC>,<VPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 14-4).
8	UPWM	R/W	U-phase PWM output setting 0: High/Low Output 1: PWM Output Port output is controlled by the combination of <UOC>,<UPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 14-4).
7-6	-	R	Read as 0.
5-4	WOC[1:0]	R/W	W-phase conduction control setting Port output is controlled by the combination of <WOC>,<WPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 14-4).
3-2	VOC[1:0]	R/W	V-phase conduction control setting Port output is controlled by the combination of <VOC>,<VPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 14-4).
1-0	UOC[1:0]	R/W	U-phase conduction control setting Port output is controlled by the combination of <UOC>,<UPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 14-4).

Note 1: Read value is the first buffer value (the latest data set via a bus).
Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
Note 3: Since the conduction control register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-3 Update Timing of the PMD1MDOUT buffer".

Table 14-4 Port Outputs according to the <UOC>,<VOC>,<WOC>,<UPWM>,<VPWM> and <WPWM> settings

PMD1MDCR<SYNTMD>="0"

Polarity: Active high (PMD1MDPOT<POLH><POLL>="11")

PMD1MDOUT Conduction Control		<WPWM><VPWM><UPWM> PWM output setting			
(Upper phase)	(Lower phase)	0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMD1MDCR<SYNTMD>=0

Polarity: Active low (PMD1MDPOT<POLH><POLL>="00")

PMD1MDOUT Conduction Control		<WPWM><VPWM><UPWM> PWM output setting			
(Upper phase)	(Lower phase)	0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	$\overline{PWM}$
0	1	H	L	H	$\overline{PWM}$
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

PMD1MDCR<SYNTMD>=1

Polarity: Active high (PMD1MDPOT<POLH><POLL>="11")

PMD1MDOUT Conduction Control		<WPWM><VPWM><UPWM> PWM output setting			
(Upper phase)	(Lower phase)	0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	$\overline{PWM}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMD1MDCR<SYNTMD>=1

Polarity: Active low (PMD1MDPOT<POLH><POLL>="00")

PMD1MDOUT Conduction Control		<WPWM><VPWM><UPWM> PWM output setting			
(Upper phase)	(Lower phase)	0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	$\overline{PWM}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

14.3.5 Protection Control Circuit

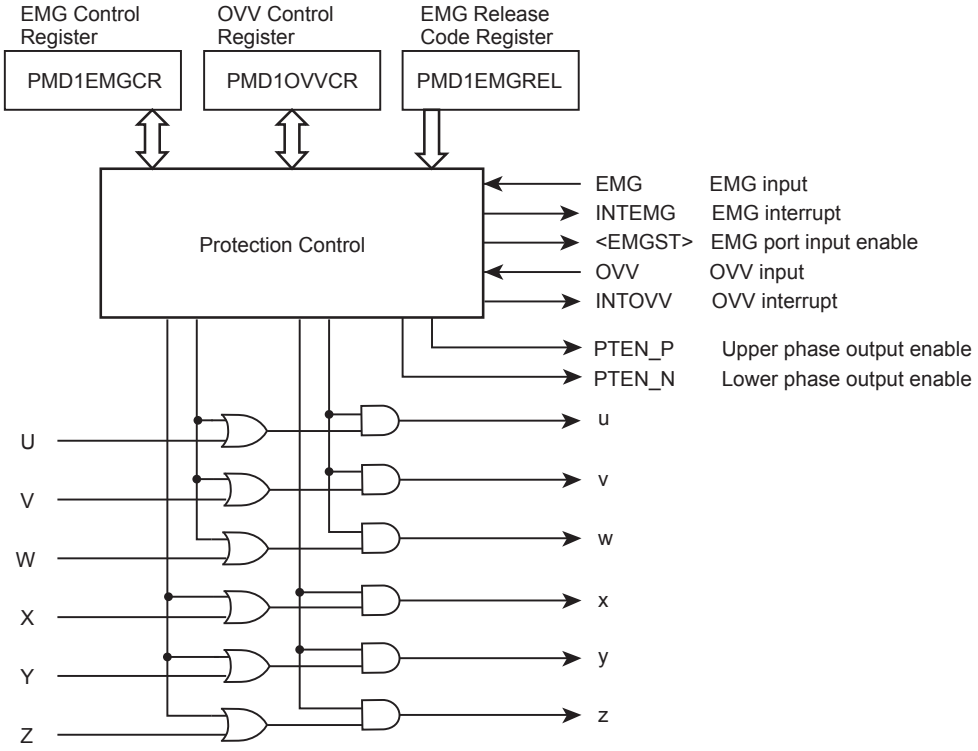


Figure 14-8 Protection Control Circuit

The protection control circuit consists of an EMG protection control circuit and an OVV protection control circuit.

14.3.5.1 EMG Protection Circuit

The EMG protection circuit consists of an EMG protection control unit and a port output disable unit. This circuit is activated when the EMG input becomes low.

The EMG protection circuit offers an emergency stop mechanism: when the EMG input is asserted (H→L), all six port outputs are immediately disabled (depending on the PMD1EMGCR<EMGMD> setting) and an EMG interrupt (INTEMG) is generated. <EMGMD> can be set to output a control signal that sets external output ports to High-z in case of an emergency.

A tool break also disables all six PWM output lines depending on the PMD1PORTMD<PORTMD> setting. When a tool break occurs, external output ports can be set to High-z through the setting of the PMD1PORTMD<PORTMD> register. A read value of 1 in EMGSTA<EMGST> indicates that the EMG protection circuit is active.

EMG protection is set through the EMG Control Register (PMD1EMGCR).

In the EMG protection state, it can be released by setting all the port output lines inactive (Set "0" to PMD1MDOUT<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC>.) (Note1) and then setting either PMD1EMGCR<EMGRS> to "1". To disable the EMG protec-tion function, write "0x5A" and "0xA5" in this order to the PMD1EMGREL register and then clear PMD1EMGCR<EMGEN> to "0". (These three instructions must be executed consecutively.) While the EMG protection input is low, any attempt to release the EMG protection state is ignored. The EMG protec-tion state can release after that confirming the status flag of PMD1EMGSTA<EMGI> is "1".

The EMG protection circuit can be disabled only after the specified key codes ("0x5A", "0xA5") are written in the PMD1EMGREL register to prevent it from being inadvertently disabled.

Note1: The data of PMD1MDOUTis necessary to be reflected in the subsequent stage buffer.

Note2: Initial procedure for EMG function

After reset, the EMG function is enabled but EMG pin is configured as a normal port. Therefore, as the EMG protection might be valid, release the EMG protection by the following procedure at the initial sequence.

1: Selects EMG function by PxFR register.

2: Reads PMD1EMGSTA<EMGI> to confirm it as "1".

3: Sets PMD1MDOUT<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC> to "0" to make all ports inactive ("L" output).

4: Releases EMG protection by setting PMD1EMGCR<EMGRS> to "1".

If the EMG protection is to be disabled, continue the following procedure.

5: Writes the key codes to PMD1EMGREL (In order of "0x5A" and "0xA5")

6: Sets PMD1EMGCR<EMGEN> to "0" to disable the EMG protection.

14.3.5.2 PMD1EMGREL (EMG Release Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	EMGREL[7:0]	W	EMG/OVV disable code The EMG and OVV protection functions can be disabled by setting 0x5A and 0xA5 in this order to register. After writing disable code, set immediately PMD1EMGCR<EMGEN>="0" or PMD1OVVCR<OVVEN>="0". When disabling these functions, <EMGEN> and <OVVEN> must be cleared to "0".

Note: Write a disable code each at disabling EMG and OVV.

14.3.5.3 PMD1EMGCR (EMG Control Register)

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

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as 0.
11-8	EMGCNT[3:0]	R/W	EMG input detection time 0x0 through 0xF (When <EMGCNT[3:0]>="0", the noise filter is bypassed.) The noise filtering length of anomaly detection input set to these bits. And this value can be calculated by following formula. <EMGCNT[3:0]> × 16/fsys (resolution: 400[nsec] at 40MHz)
7-6	-	R	Read as 0.
5	INHEN	R/W	Tool break enable/disable 0: Disable 1: Enable This bit selects whether or not to stop the PMD when the PMD stop signal is input from the tool. Note: Tool break is enabled in the initial status.
4-3	EMGMD[1:0]	R/W	EMG protection mode select 00: All phases High-Z 01: All upper-phase ON / all lower-phase High-Z 10: All upper phase High-Z / all lower phase ON 11: All phase High-Z Sets the port output both the upper (UO, VO, WO) and the lower (XO, YO, ZO) for the case when EMG occurs. Note: "ON" indicates that PWM output continues.
2	-	R/W	Always write "0".
1	EMGRS	W	EMG protection release 0: - 1: Release protection EMG protection can be released by setting the PMD1MDOUT register to "0x000" and then setting the <EMGRS> bit to "1". Note: This bit is always read as 0. Note: EMG protection cannot be released if the subsequent stage buffer of PMD1MDOUT is not updated to "0x000". Note: Before releasing EMG protection, make sure that the PMD1EMGSTA<EMGI> has returned to "1".
0	EMGEN	R/W	EMG protection circuit enable/disable 0: Disable 1: Enable To disable the function, write "0x5A" and then write "0xA5" to the EMG release register(PMD1EMGREL). Then, set "0" to <EMGEN> (Thse three instructions must be executed consecutively.). Note: This EMG protection circuit is enabled in the initial status.

14.3.5.4 PMD1EMGSTA (EMG Status Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1	EMGI	R	EMG input EMG protection state The EMG input state can be distinguished by reading this bit
0	EMGST	R	EMG protection state 0: Normal operation 1: Protected The EMG protection state can be distinguished by reading this bit.

14.3.5.5 OVV Protection Control Circuit (OVV Input Block)

The OVV protection control circuit consists of an OVV protection control unit and a port output disable unit. This circuit is activated when the OVV input port is asserted.

When the OVV input signal is asserted (H→L) for a specified period (set to PMD1OVVCR <OVVCNT>), the OVV protection circuit fixes the six port output lines in the conduction control circuit to high or low. At this time, an OVV interrupt (INTOVV) is generated. It is possible to select only the upper phase, lower phase or all phase.

OVV protection is set through the PMD1OVVCR of OVV control register. A read value of "1" in PMD1OVVSTA<OVVST> indicates that the OVV protection circuit is active.

The release of the OVV protection state is enabled by setting PMD1OVVCR<OVVRS> to "1". And after the protection input is canceled, OVV protection is automatically released at a predetermined timing. (The OVV protection state is not released while the OVV protection input is low. The state of this port input can be checked by reading PMD1OVVSTA<OVVI>.)

The OVV protection state is released in synchronization with the PWM period (at the timing when PWM count PMD1MDCNT matches PMD1MDPRD. However if an interrupt on a half cycle of PWM is set, the protection state is released when PWM count is "1" or matches PMD1MDPRD.). To disable the OVV protection function, write "0x5A" and "0xA5" in this order to the PMD1EMGREL of EMG release register and then clear PMD1OVVCR<OVVEN> to "0". (These three instructions must be executed consecutively.)

The OVV protection circuit can be disabled only after the specified key codes ("0x5A", "0xA5") are written in the PMD1EMGREL register to prevent it from being inadvertently disabled.

14.3.5.6 PMD1OVVCR (OVV 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	-	-	-	-	OVVCNT			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as 0.
11-8	OVVCNT[3:0]	R/W	OVV input detection time Value: 0x1 through 0xF (If "0" is set, it is handled as "1".) The noise filtering length of OVV input set to these bits. And this value can be calculated by following formula. <OVVCNT[3:0]> × 16/fsys (resolution: 400[nsec] at 40MHz) Note: <OVVCNT[3:0]> is effective only when port input is selected (<OVVISEL>="0").
7	-	R	Read as 0.
6	ADIN1EN	R/W	ADC monitoring function 1 input enable 0: Disable input 1: Enable input Selects enable/disable signals from ADC monitoring function 1 of the AD converter. If you enable it and select ADC monitoring signal for input (<OVVISEL>="1"), the results of the ADC monitoring function 1 as OVV inputs (if OVV protection is enabled). Note: Refer to the chapter "AD conversion monitoring function" in the "12-bit analog/digital conveter" for detailed information about AD conversion monitoring function.
5	ADIN0EN	R/W	ADC monitoring function 0 input enable 0: Disable input 1: Enable input Selects enable/disable signals from ADC monitoring function 0 of the AD converter. If you enable it and select ADC monitoring signal for input (<OVVISEL>="1"), the results of the ADC monitoring function 0 as OVV inputs (if OVV protection is enabled). Note: Refer to the chapter "AD conversion monitoring function" in the "12-bit analog/digital conveter" for detailed information about AD conversion monitoring function.
4-3	OVVMD[1:0]	R/W	Selects OVV protection mode 00: No output control 01: All upper phase ON, all lower phase OFF 10: All upper phase OFF, all lower phase ON 11: All phase OFF This field controls the outputs of the upper (UO,VO,WO) and lower(XO,YO,ZO) phases when an OVV condition occurs. Note: "ON" indicates that it's fixed to active output. "OFF" indicates that it's fixed inactive output. Active and inactive are depends on the settings of <POLL> and <POLH>. Note: If OVV and EMG conditions occur simultaneously, the protection mode settings in the bits of <EMGMD[1:0]> become effective.
2	OVVISEL	R/W	Selects OVV input 0: Port input 1: ADC monitor signal This bit selects whether to use port input or the monitor signal from the ADC as the OVV signal to be input to the protection circuit. Note: When the ADC monitor signal is selected, the setting of OVV input detection time <OVVCNT[3:0]> becomes invalid.(Direct input)
1	OVVRS	R/W	Selects OVV protection state release 0: Disable automatic release of OVV protection state 1: Enable automatic release of OVV protection state Note: If automatic release of OVV protection is enabled, when the state changes to OVV protection after detecting anomaly (OVV input makes a high-to-low transition), the OVV protection state can be automatically released when updating the buffer of PWM cycle register PMD1MDPRD after the OVV input transition to "high". (Refer to the "Table 14-2 PMD1MDPRD, PMD1CMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing")
0	OVVEN	R/W	OVV protection circuit enable/disable 0: Disable 1: Enable Note: To disable the function, write "0x5A" and then write "0xA5" to the EMG release register (PMD1EMGREL). Then, set "0" to <OWEN>. (These three instructions must be executed consecutively.)

14.3.5.7 PMD1OVVSTA (OVV 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	-	-	-	-	-	-	OVVI	OVVST
After reset	0	0	0	0	0	0	Undefined	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1	OVVI	R	OVVI input OVVI state The OVV input state (selected by PMD1OVVCR<OVVISEL>) can be distinguished by reading this bit.
0	OVVST	R	OVV protection state 0: Normal operation 1: In protected The OVV state can be distinguished by reading this bit.

14.3.6 Dead Time Circuit

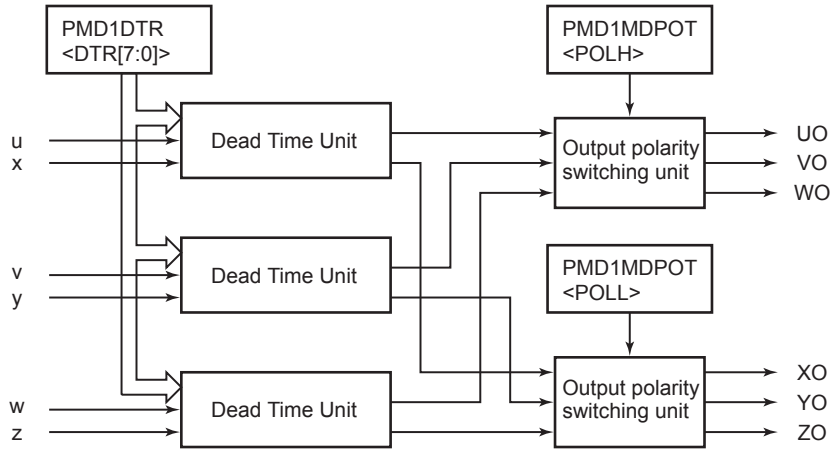


Figure 14-9 Dead Time Circuit

The dead time circuit consists of a dead time unit and an output polarity switching unit.

For each of the U, V and W phases, the dead time units delay the ON-timing of each phase when the upper and lower phases are switched to prevent a short circuit. The dead time is set to the Dead Time Register (PMD1DTR<DTR[7:0]>) as an 8-bit value with a resolution of 200[ns] at 40MHz can be set..

The output polarity switching circuit allows the polarity (active high or active low) of the upper-output (UO, VO, WO) and lower-output (XO,YO,ZO) phases to be independently set through PMD output setting register PMD1MDPOT<POLH> and <POLL>.

14.3.6.1 PMD1DTR (Dead Time Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DTR[7:0]	R/W	Sets Dead time 0x00 through 0xFF The Dead time value can be calculated by following formula. 200 nsec × <DTR[7:0]> (up to 51 µsec at fsys = 40 MHz)

14.3.7 Sync Trigger Generation Circuit

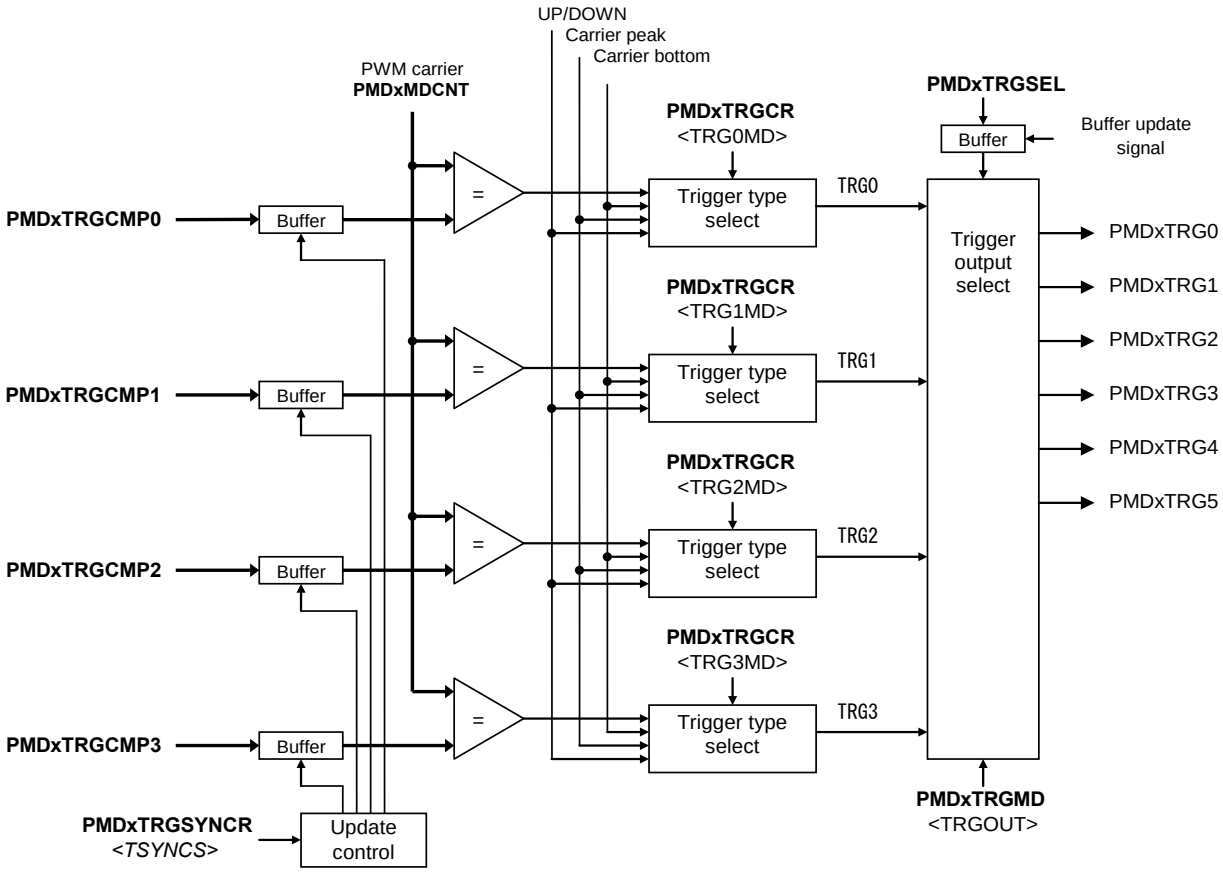


Figure 14-10 Sync Trigger Generation Circuit

The sync trigger generation circuit generates four trigger signals (TRG0 to TRG3) for starting ADC sampling in synchronization with PWM.

The trigger timing can be selected following 6 types.

1. At up count operation compare-match (Note)
2. At down count operation compare-match (Note)
3. At up-/down count operation compare-match (Note)
4. PWM carrier peak
5. PWM carrier bottom
6. PWM carrier peak and PWM carrier bottom

Note: The compare-match is between PWM counter (PMD1MDCNT<MDCNT[15:0]>) and (PMD1TRGCMPn <TRGCMPn[15:0]>)

During in trigger select output mode: PMD1TRGMD<TRGOUT>="1". The TRG0 signal is output from PMD1TRG0 to 5 selected by the trigger output select register PMD1TRGSEL. The TRG0 setting is set by PMD1TRGCMP0 and PMD1TRGCR<TRG0MD>.

When the edge mode (sawtooth wave carrier mode) is selected, the compare-match function is up count. When PMD1TRGMD<EMGTGE>="1", this circuit also outputs trigger signals in EMG protection state.

14.3.7.1 PMD1TRGCMP0 (Trigger Compare Registers 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	TRGCMP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP0 [15:0]	R/W	Trigger output compare register <TRGCMP0[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP0. TRG0 is output. Note: It is prohibited to set <TRGCMP0> to "0" and <TRGCMP0> ≥ <MDPRD[15:0]> value.

Note 1: Read value is the first buffer value (the latest data set via a bus).
Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
Note 3: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-5 Buffer update timing for the trigger compare register".

Table 14-5 Buffer update timing for the trigger compare register

<TSYNCS>setting	<TRGnMD>setting	TRGCMPn register Buffer update timing
00	000	Updates immediately
	001	Update when PWM carrier peak
	010	Update when PWM carrier bottom
	011	Update when PWM carrier peak or PWM carrier bottom (Note1)
	1xx	Updates immediately
01	xxx	Update when PWM carrier bottom
10	xxx	Update when PWM carrier peak
11	xxx	Update when PWM carrier peak or PWM carrier bottom (Note1)

Note: x : Don't care

Note: Asynchronous update PMD1MDEN<PWMEN>="0" regardless of setting.

Note1: Updates at carrier peak when sawtooth wave carrier is selected (PMD1MDCR<PWMMD>="0") .

14.3.7.2 PMD1TRGCMP1 (Trigger Compare Registers1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRGCMP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP1 [15:0]	R/W	Trigger output compare register <TRGCMP1[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP1. TRG1 is output. Note: It is prohibited to set <TRGCMP1> to "0" and <TRGCMP1> ≥ <MDPRD[15:0]> value.

Note 1: Read value is the first buffer value (the latest data set via a bus).
Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
Note 3: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-5 Buffer update timing for the trigger compare register".

14.3.7.3 PMD1TRGCMP2 (Trigger Compare Registers 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	TRGCMP2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP2							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP2 [15:0]	R/W	Trigger output compare register <TRGCMP2[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP2. TRG2 is output. Note: It is prohibited to set <TRGCMP2> to "0" and <TRGCMP2> ≥ <MDPRD[15:0]> value.

Note 1: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 2: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-5 Buffer update timing for the trigger compare register".

Note 4: Read value is the first buffer value (the latest data set via a bus).

14.3.7.4 PMD1TRGCMP3 (Trigger Compare Registers 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	TRGCMP3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP3 [15:0]	R/W	Trigger output compare register <TRGCMP3[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP3. TRG3 is output. Note: It is prohibited to set <TRGCMP3> to "0" and <TRGCMP3> ≥ <MDPRD[15:0]> value.

Note 1: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 2: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: For detailed update timing of the subsequent stage buffer, refer to the "Table 14-5 Buffer update timing for the trigger compare register"

Note 4: Read value is the first buffer value (the latest data set via a bus).

14.3.7.5 PMD1TRGCR (Trigger Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15	TRG3BE	R/W	Asynchronous update of the PMD1TRGCMP3<TRGCMP3[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMD1TRGCMP3. 0: Sync update 1: Async update (The value written to PMD1TRGCMP3 is immediately reflected.) Note: For detailed update timing, refer to the "Table 14-5 Buffer update timing for the trigger compare reg-ister". Note: When PMD1MDEN<PWMEN>="0", updates asynchronously regardless of setting.
14-12	TRG3MD[2:0]	R/W	PMD1TRGCMP3 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMD1MDCR<PWMMD> (a sawtooth wave), a trigger is output on up-coun-ter match even "001" is set to <TRG3MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG3MD[2:0]>. Note: When <TRG3MD[2:0]>="011", PMD1TRGCMP3="0x0001" and PMD1MDCR<PWMMD>="1" (trian-gular wave), one trigger output is made per period.
11	TRG2BE	R/W	Asynchronous update of the PMD1TRGCMP2<TRGCMP2[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMD1TRGCMP2. 0: Sync update 1: Async update (The value written to PMD1TRGCMP2 is immediately reflected.) Note: For detailed update timing, refer to the "Table 14-5 Buffer update timing for the trigger compare reg-ister". Note: When PMD1MDEN<PWMEN>="0", updates asynchronously regardless of setting.

Bit	Bit Symbol	Type	Function
10-8	TRG2MD[2:0]	R/W	PMD1TRGCMP2 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMD1MDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG2MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG2MD[2:0]>. Note: When <TRG2MD[2:0]>="011", PMD1TRGCMP2="0x0001" and PMD1MDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
7	TRG1BE	R/W	Asynchronous update of the PMD1TRGCMP1<TRGCMP1[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMD1TRGCMP1. 0: Sync update 1: Async update (The value written to PMD1TRGCMP1 is immediately reflected.) Note: For detailed update timing, refer to the "Table 14-5 Buffer update timing for the trigger compare reg-ister". Note: When PMD1MDEN<PWMEN>="0", updates asynchronously regardless of setting.
6-4	TRG1MD[2:0]	R/W	PMD1TRGCMP1 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMD1MDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG1MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG1MD[2:0]>. Note: When <TRG1MD[2:0]>="011", PMD1TRGCMP1="0x0001" and PMD1MDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
3	TRG0BE	R/W	Asynchronous update of the PMD1TRGCMP0<TRGCMP0[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMD1TRGCMP0. 0: Sync update 1: Async update (The value written to PMD1TRGCMP0 is immediately reflected.) Note: For detailed update timing, refer to the "Table 14-5 Buffer update timing for the trigger compare reg-ister". Note: When PMD1MDEN<PWMEN>="0", updates asynchronously regardless of setting.

Bit	Bit Symbol	Type	Function
2-0	TRG0MD[2:0]	R/W	PMD1TRGCMP0 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMD1MDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG0MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG0MD[2:0]>. Note: When <TRG0MD[2:0]>="011", PMD1TRGCMP0="0x0001" and PMD1MDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.

14.3.7.6 PMD1TRGSYNCR (Trigger Update Timing 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	-	-	-	-	-	-	TSYNCS	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1-0	TSYNCS	R/W	Update timing setting for the buffer of the trigger compare register. 00: Updates immediately, Update when PWM carrier peak, bottom and peak or bottom is set for each trigger by setting PMD1TRGCR<TRGxMD>. 01: Update when PWM carrier bottom 10: Update when PWM carrier peak 11: Update when PWM carrier peak or bottom Note: Refer to the "Table 14-5 Buffer update timing for the trigger compare register". Note: Asynchronous update PMD1MDEN<PWMEN>="0" regardless of setting.

14.3.7.7 PMD1TRGMD (Trigger Output Mode 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	-	-	-	-	-	-	TRGOUT	EMGTGE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1	TRGOUT	R/W	Trigger output mode 0: Fixed trigger output 1: Variable trigger output When fixed trigger outputs are selected, trigger outputs from PMD1TRG0 to PMD1TRG3 output the trigger signals generated by a match with <TRGCMP0[15:0]> to <TRGCMP3[15:0]> respectively. A PMD1TRG4 and a PMD1TRG5 are not output the trigger signals. When variable trigger output is selected, output signals of the <TRGCMP0[15:0]> are output to one of trigger output from PMD1TRG0 through PMD1TRG5. The trriger output signal is selected by trigger output select register. Note: Refer to the "Table 14-6 Trigger Output Patterns" when variable trigger outputs is selected (<TRGOUT>="1").
0	EMGTGE	R/W	Output enable in EMG protection state 0: Disable trigger output in the protection state 1: Enable trigger output in the protection state This bit enables or disables trigger output in the EMG protection state.

Table 14-6 Trigger Output Patterns

<TRGOUT> Setting	Compare Register	<TRGSEL[2:0]> Setting	Trigger Output
<TRGOUT>="0"	PMD1TRGCMP0	×	PMD1TRG0
	PMD1TRGCMP1		PMD1TRG1
	PMD1TRGCMP2		PMD1TRG2
	PMD1TRGCMP3		PMD1TRG3
<TRGOUT>="1"	PMD1TRGCMP0	0	PMD1TRG0
		1	PMD1TRG1
		2	PMD1TRG2
		3	PMD1TRG3
		4	PMD1TRG4
		5	PMD1TRG5
	PMD1TRGCMP1	×	No trigger output
	PMD1TRGCMP2	×	No trigger output
	PMD1TRGCMP3	×	No trigger output

14.3.7.8 PMD1TRGSEL (Trigger Output Select Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
2-0	TRGSEL[2:0]	R/W	Trigger output select 000: Output from PMD1TRG0 001: Output from PMD1TRG1 010: Output from PMD1TRG2 011: Output from PMD1TRG3 100: Output from PMD1TRG4 101: Output from PMD1TRG5 110: No trigger output 111: No trigger output This field is effective when the variable trigger output mode is selected (PMD1TRGMD<TRGOUT>="1"). And an output trigger can be selected by setting the PMD1TRGCMP0 register. (Refer to the Table 14-6.)

Note 1: When PMD is disabled (PMD1MDCR<PWMEN>="0"), updates asynchronously.
Note 2: Since the trigger output selecting register is double buffered, the cycle can be changed during the operation of the PWM counter.
Note 3: The update timing of the subsequent stage buffer is as the same as the compare register (PMD1CMPU/V/W).

15. Encoder Input Circuit (ENC)

15.1 Outline

The encoder input circuit supports four operation modes including encoder mode, sensor mode (two types) and timer mode. And the functions are as follows:

- Supports incremental encoders and Hall sensor ICs. (signals of Hall sensor IC can be input directly)
- 24-bit general-purpose timer mode
- Multiply-by-4 (multiply-by-6) circuit
- Rotational direction detection circuit
- 24-bit counter
- Comparator enable/disable
- Interrupt request output:1
- Digital noise filters for input signals

15.2 Differences between channels

The TMPM3U0FSDMG has a one-channel incremental encoder interface (ENC0), which can obtain the absolute position of the motor, based on input signals from the incremental encoder.

These channels operate identical except the differences in below.

Table 15-1 Differences between channels

Channel	Input pin			Encoder input interrupt
	A-phase	B-phase	Z-phase	
Channel0	ENCA0	ENCB0	ENCZ0	INTENC0

15.3 Block Diagram

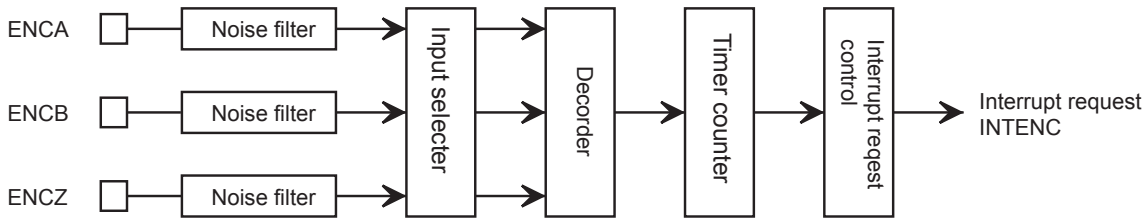


Figure 15-1 Block diagram of encoder input circuit

15.4 Registers

15.4.1 List of Registers

The following is control registers and addresses of encoder input circuit.

Channel x	Base Address
Channel0	0x4001_0400

Register name		Address(Base+)
Encoder Input Control Register	ENTNCR	0x0000
Encoder Counter Reload Register	ENRELOAD	0x0004
Encoder Compare Register	ENINT	0x0008
Encoder Counter	ENCNT	0x000C

15.4.2 ENTNCR(Encoder Input Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	MODE		P3EN
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ZEN	ENRUN	NR		INTEN	ENDEV		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	–	R	Read as "0".
18-17	MODE[1:0]	R/W	Encoder input mode setting 00:Encoder mode 01:Sensor mode (event count)) 10:Sensor mode (timer count)) 11:Timer mode
16	P3EN	R/W	2-phase / 3-phase input selection (sensor mode) (Note 1) 0:2-phase input 1:3-phase input Sets the number of input signals.
15	CMP	R	Compare flag 0:- 1:Compare (Clear by RD) If comparing is executed, <CMP> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect.
14	REVERR	R	Reverse error flag (Sensor mode (at timer count)) (Note 2) 0:- 1:Error (Clear by RD) In sensor mode (at timer count), when a reverse error occurs, <REVERR> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect. In the encoder mode, sensor mode (event count) and timer mode, this bit has no meaning.
13	UD	R	Rotation direction 0:CCW direction (Counter-clockwise) 1:CW direction (Clockwise) <UD> is set to "0", when <ENRUN> = "0".
12	ZDET	R	Z-Detected 0:Not detected 1:Z-phase detected <ZDET> is set to 1 on the first edge of Z input signal (ENCZ) after <ENRUN> is written from 0 to 1. This occurs on a rising edge of the signal Z during CW rotation or on a falling edge of Z during CCW rotation. <ZDET> is set to "0" when <ENRUN> = "0". <ZEN> has no influence on the value of <ZDET>. <ZDET> is set to "0" in the sensor event count and the sensor timer count modes. In the sensor mode (event count) and sensor mode (timer count), this bit is always set to "0".

Bit	Bit Symbol	Type	Function
11	SFTCAP	W	Executes software capture (timer mode/sensor mode (at timer count)) 0:- 1:Software capture If <SFTCAP> is set to 1, the value of the encoder counter is captured into the ENCNT register. Writing "0" to <SFTCAP> has no effect. Reading <SFTCAP> always returns to "0". In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; writing "1" to this bit is ignored.
10	ENCLR	W	Encoder pulse counter clear 0:- 1:Clear Writing a 1 to <ENCLR> clears the encoder counter to "0". Once cleared, the encoder counter restarts counting from 0. Writing "0" to <ENCLR> has no effect. Reading <ENCLR> always returns to "0".
9	ZESEL	R/W	Edge selection of ENCZ (timer mode) 0:Rising edge 1:Falling edge In timer mode, this bit selects inputs edge of ENCZ used as external trigger. In the other mode, this bit has no meaning.
8	CMPEN	R/W	Compare enable 0:Disable 1:Enable When "1" is set to <CMPEN>, this bit compares counter values of encoder counter with register value of ENINT. When "0" is set to <CMPEN> , this compare is disabled.
7	ZEN	R/W	Z-phase enable (Encoder mode/timer mode) 0:Disable 1:Enable In the other mode, this bit has no meaning | <Encoder mode> Clear setting of encoder counter using ENCZ input | When <ZEN> = "1" is set, if a rising edge of ENCZ is detected during rotating clockwise, the encoder counter is cleared to "0". If a falling edge of ENCZ is detected during rotating counter- clockwise, the encoder counter is cleared to "0". If the edges of ENCLK (multiply by 4 clock derived from the decoded A and B signals) and the edge of ENCZ coincide, the encoder counter is cleared to "0" without incrementing or decrementing (i.e., the clear takes precedence). | | <Timer mode> Sets ENCZ input to use as an external trigger. | When <ZEN> = 1, the value of the encoder counter is captured into the ENINT register and cleared to "0" on the edge of ENCZ selected by <ZESEL>. |
6	ENRUN	R/W	Encoder operation enable 0:Disable 1:Enable Setting <ENRUN> to 1 and clearing <ZDET> to "0" enables the encoder operation. Clearing <ENRUN> to "0" disables the encoder operation. There are counters and flags that are cleared and not cleared when <ENRUN> bit is cleared to "0".
5-4	NR[1:0]	R/W	Noise filter 00:No filtering 01:Filters out pulses narrower than 31/fsys as noise 10:Filters out pulses narrower than 63/fsys as noise 11:Filters out pulses narrower than 127/fsys as noise The digital noise filters remove pulses narrower than the width selected by <NR[1:0]>.
3	INTEN	R/W	Encoder interrupt enable 0:Disable 1:Enable <INTEN> enables or disables the ENC interrupt. Setting <INTEN> to "1" enables interrupt generation. Setting <INTEN> to "0" disables interrupt generation.

Bit	Bit Symbol	Type	Function
2-0	ENDEV[2:0]	R/W	Encoder pulse division factor 000:divided by 1 100:divided by 16 001:divided by 2 101:divided by 32 010:divided by 4 110:divided by 64 011:divided by 8 111:divided by 128 Sets encoder pulse division factor The frequency of the encoder pulse is divided by the factor specified by <ENDEV[2:0]>. The divided signal determines the interval of the event interrupt.

Note 1: In the encoder mode or timer mode, <P3EN> must be set to "0".
Note 2: If changing the mode, first read the flag to clear.

The operation mode has eight modes specified with<MODE[1:0]>, <P3EN> and <ZEN>.

The operation mode settings are as follows:

<MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Mode
00	0	0	A, B	Encoder mode
	1		A,B,Z	Encoder mode (use of Z)
01	0	0	U,V	Sensor mode (event count, 2-phase input)
		1	U,V,W	Sensor mode (event count, 3-phase input)
10	0	0	U,V	Sensor mode (timer count, 2-phase input)
		1	U,V,W	Sensor mode (timer count, 3-phase input)
11	0	0	-	Timer mode
	1		Z	Timer mode (use of Z)

The following is the status of <ENRUN> and corresponding signals.

Counter/flag	<ENRUN> = 0 (After reset)	<ENRUN> = 1 (Operating)	<ENRUN> = 0 (Stopping)	<ENRUN> = 0 Object flag/counter clear procedure
Encoder counter	0x000000	Count operation	Maintains a value when stopping	Software clear (<ENCLR> = 1 WR)
Noise filter counter	0y0000000	Count-up operation	Count-up operation (Always filtering)	Only reset
Encoder pulse division counter	0x00	Count-down operation	Stopped and cleared	Clear when <ENRUN> = "0"
Compare flag <CMP>	0	"1" is set when comparing Clear when read.	Cleared	Clear when <ENRUN> = "0"
Reverse error flag <REVERR>	0	"1" is set when error occurs. Clear when read.	Cleared	Clear when <ENRUN> = "0"
Z detection flag <ZDET>	0	"1" is set when Z is detected.	Cleared	Clear when <ENRUN> = "0"
Rotation direction bit <UD>	0	"0"/"1" is set depending on the direction	Cleared	Clear when <ENRUN> = "0"

15.4.3 ENRELOAD(Encoder Counter Reload Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	RELOAD[15:0]	R/W	Sets the Encoder counter period (after multiplied by 4 or 6) 0x0000 to 0xFFFF Z-phase is used : Sets the number of count pulses for one rotation Z-phase is not used : Sets the number of count pulses minus one for one rotation <RELOAD[15:0]> defines the encoder counter period multiplied by 4. If the encoder counter is configured as an up-counter, it increments up to the value programmed in <RELOAD[15:0]> and then wraps around to "0" on the next ENCLK. If the encoder counter is configured as a down-counter, it decrements to "0" and then is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.

Note 1: The RELOAD register is only used in Encoder mode.

Note 2: ENRELOAD register should be accessed with 32-bit instructions.

15.4.4 ENINT(Encoder Compare Register)

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

Bit	Bit Symbol	Type	Function		
31-24	−	R	Read as "0".		
23-0	INT[23:0]	R/W	Counter compare value setting		
			Encoder mode:	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = "1" is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = "1"is set, an interrupt request (INTENC0) occurs. However if <ZEN> = "1" is set, an interrupt request does not occur until <ZDET> = "1".	
			Sensor mode: (event count)	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = "1" is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = "1"is set, an interrupt request (INTENC0) occurs. This bit has no effect on a value of <ZEN>.	
			Sensor mode: (Timer count)	Interrupt condition of abnormal pulse detection time	0x000000 to 0xFFFFFF
				When <CMPEN> = "1" is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = "1" is set, an interrupt request (INTENC0) occurs. This bit has no effect on a value of <ZEN>.	
			Timer mode	Interrupt condition of timer compare	0x000000 to 0xFFFFFF
				When <CMPEN> = "1" is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = "1" is set, an interrupt request (INTENC0) occurs. This bit has no effect on a value of <ZEN>.	

Note 1: <INT[23:16]> is used only in Sensor mode (timer count) and Timer mode.

Note 2: ENINT register should be accessed with 32-bit instructions.

15.4.5 ENCNT (Encoder Counter)

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

Bit	Bit Symbol	Type	Function		
31-24	−	R	Read as "0".		
23-0	CNT[23:0]	R/W	Encoder counter/capture value		
			Encoder mode:	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Encoder mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to the value of <RELOAD[15:0]>, it wraps around to "0" on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to "0", it is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.	
			Sensor mode: (event count)	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Sensor Event Count mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to "0xFFFF", it wraps around to "0" on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to "0", it wraps around to "0xFFFF" on the next ENCLK.	
			Sensor mode: (Timer count)	Pulse detection time or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read. In Sensor mode, the value of encoder counter can be read and captured by software on each encoder pulse (ENCLK) by writing "1" to <SFTCAP>. The captured value is cleared to "0" by system reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Sensor Timer Count mode, the encoder counter is configured as a free-running counter that counts up with fsys. The encoder counter is cleared to "0" when the encoder pulse (ENCLK) is detected. When it has reached to "0xFFFFFF", it wraps around to "0" automatically.	
			Timer mode	Capture value of internal counter or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read and captured by software by writing "1" to <SFTCAP>.When <ZEN> = "1", the value of the encoder counter is also captured into <CNT23:0> on the Z edge selected by <ZESEL>. The captured value is cleared to "0" by reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached to "0xFFFFFF", it wraps around to "0" automatically.	

Note 1: <CNT[23:16]> is used only in the sensor mode (Timer counting) or timer mode. In the encoder mode or sensor mode (event counting), always reads as "0".

Note 2: ENCNT register should be accessed with 32-bit instructions.

15.5 Operational Description

15.5.1 Encoder mode

The high-speed position sensor determines the phase input from the AB encoder and the ABZ encoder.

- Event detection (rotation pulse) → interrupt generation
- Event count → match detection interrupt generation (measures the amount of transferring)
- Detects rotation direction
- Up/down-count (changeable in operation)
- Settable counter cycle

15.5.2 Sensor mode

The low-speed position sensor determines (zero-cross determination) the phase input from UV Hall sensor and UVW Hall sensor.

There are two kinds of sensor modes such as event count mode and timer count mode (counts with fsys).

15.5.2.1 Event Count Mode

- Event detection (rotation pulse) → interrupt generation
- Event count → match interrupt occurs (measuring the amount of transfer)
- Rotation direction detection

15.5.2.2 Timer count mode

- Event detection (rotation pulse) → interrupt generation
- Timer count
- Rotation direction detection
- Capture function → event capture (measures event intervals) → interrupt generation

  software capture
- Abnormal detection time error (timer compare) → match detection interrupt generation
- Reverse detection error → error flag caused by changing rotation direction

15.5.3 Timer mode

This mode can be used as a general-purpose 24-bit timer.

- 24-bit up counter
- Counter clear control (software clear, timer clear, external trigger and free-run count)
- Compare function → match detection interrupt generation
- Capture function → external trigger capture → interrupt generation

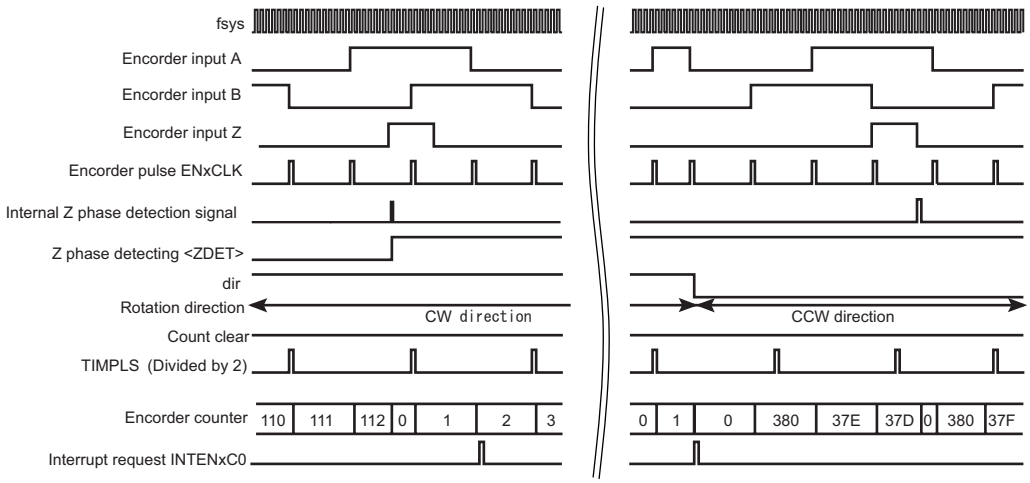
  software capture

15.6 Function

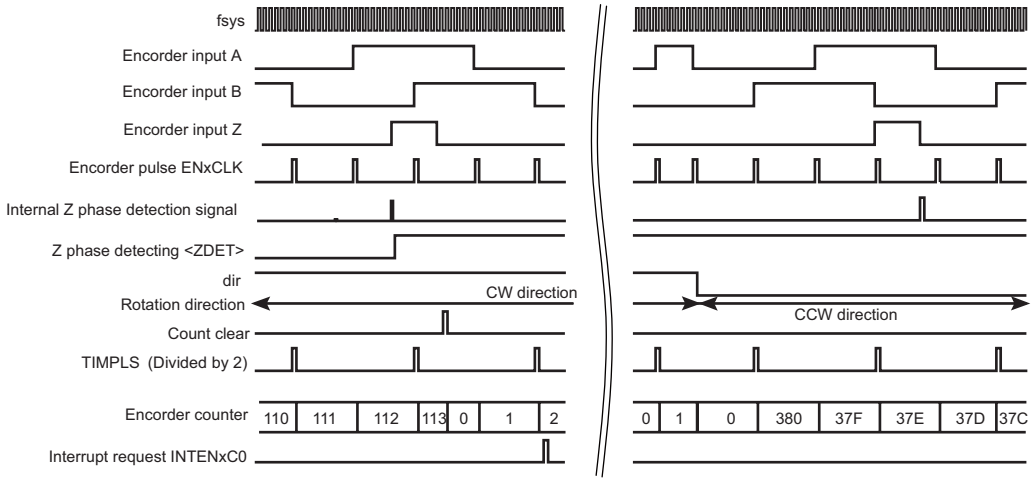
15.6.1 Mode operation outline

15.6.1.1 Encoder mode

1. If <ZEN> = 1 (<RELOAD> = 0x0380 , <ENINT> = 0x0002)



2. If <ZEN> = 0 (<RELOAD> = 0x0380 , <ENINT> = 0x0002)

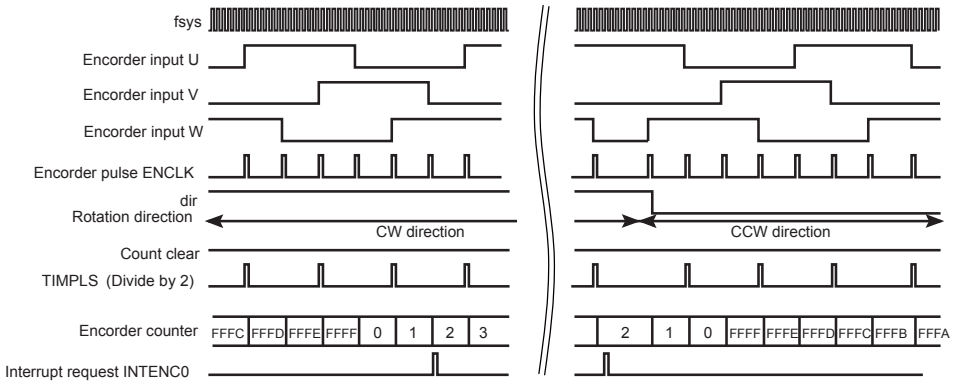


- The incremental encoder inputs of the MCU should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied by 4 clock derived from the decoded A and B quadrature signals.
- During CW rotation (i.e., A has the 90-degree phase lead to B), the encoder counter counts up; when it has reached to the value of <RELOAD>, it wraps around to "0" on the next ENCLK.
- During CCW rotation (i.e., A has the 90-degree phase lag to B), the encoder counter counts down; when it has reached to "0x0000", it is reloaded with the value of <RELOAD> on the next ENCLK.

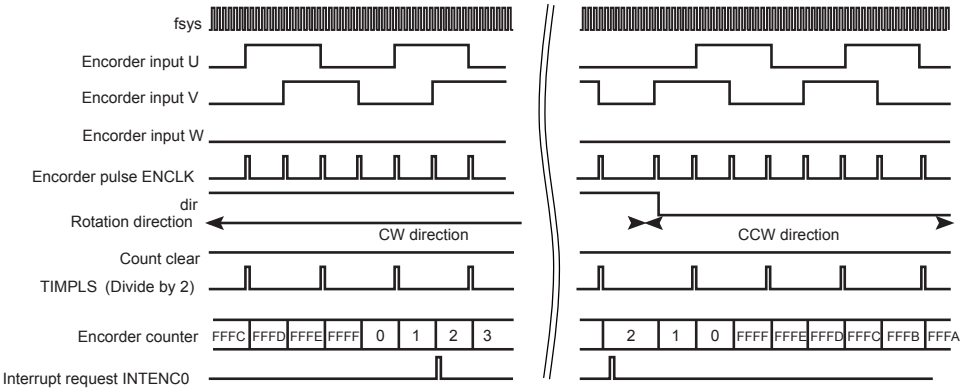
- Additionally, when <ZEN> = "1", the encoder counter is cleared to "0" on the rising edge of Z during CW rotation and on the falling edge of Z during CCW rotation (at the internal Z_Detected timing). If the ENCLK edge matches Z edge, the encoder counter is cleared to "0" without incrementing or decrementing.
- When <ENCLR> is set to 1, the encoder counter is cleared to "0".
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of <ENINT>. When <ZEN> = "1", however, an interrupt does not occur while <ZDET> = "0".
- When <ZDET> and <UD> are set to "0", <ENRUN> is cleared to "0".

15.6.1.2 Sensor mode (event count)

1. If <P3EN> = 1 (<ENINT> = 0x0002)



2. If <P3EN> = 0 (<ENINT> = 0x0002)

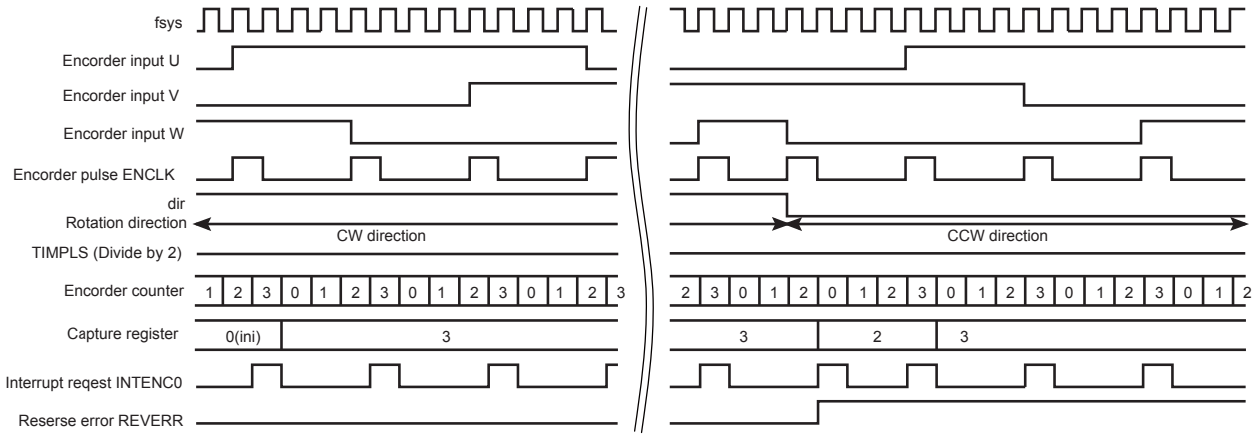


- The Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter counts the pulses of ENCLK, which is either multiplied by 4 clock (when <P3EN> = "0") derived from the decoded U and V signals or multiplied by 6 clock (when <P3EN> = "1") derived from the decoded U, V and W signals.
- During CW rotation (i.e., U channel has the 90-degree phase lead to V channel; V channel has the 90-degree phase lead to W channel), the encoder counter counts up; when it has reached to "0xFFFF", it wraps around to "0" on the next ENCLK.

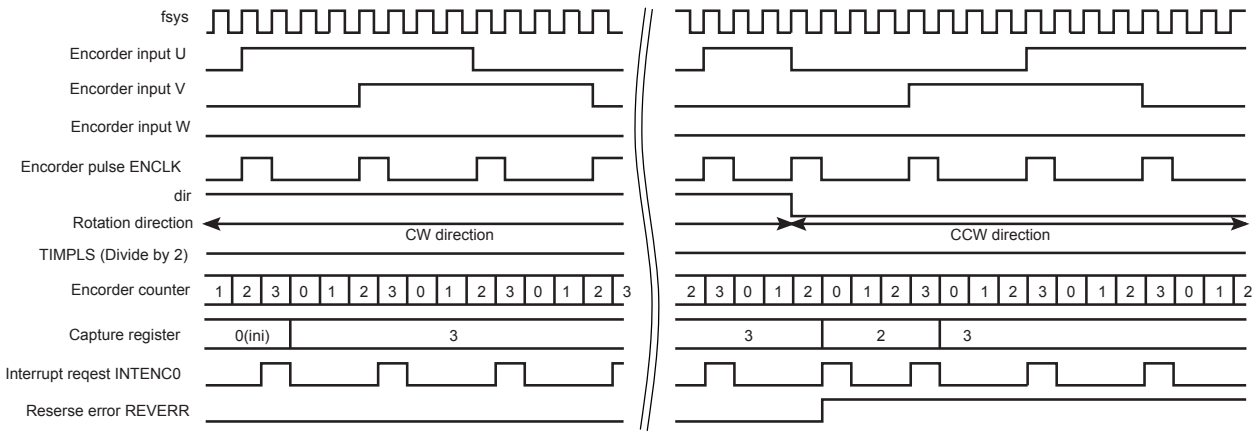
- During CCW rotation (i.e., U channel has the 90-degree phase lag to V channel; V channel has the 90-degree phase lag to W), the encoder counter counts down; when it has reached to "0x0000", it wraps around to "0xFFFF" on the next ENCLK.
- When <ENCLR> is set to 1, the internal counter is cleared to "0".
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the internal counter has reached to the value of <ENINT>.
- When <UD> and <ENRUN> are set to "0", <UD> is cleared to "0".

15.6.1.3 Sensor mode (Timer count)

1. If <P3EN> = 1 (<ENINT> = 0x0002)



2. If <P3EN> = 0 (<ENINT> = 0x0002)

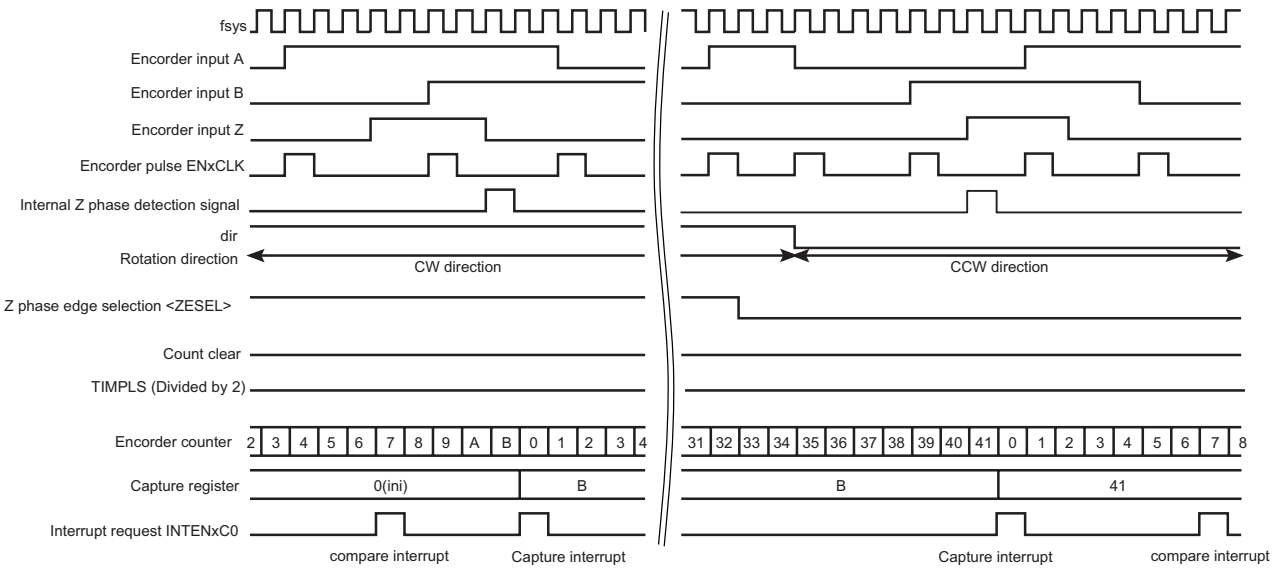


- In Sensor Timer Count mode, the Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied by 4 clock (when <P3EN> = "0") derived from the decoded U and V signals or multiplied by 6 clock (when <P3EN> = "1") derived from the decoded U, V and W signals.

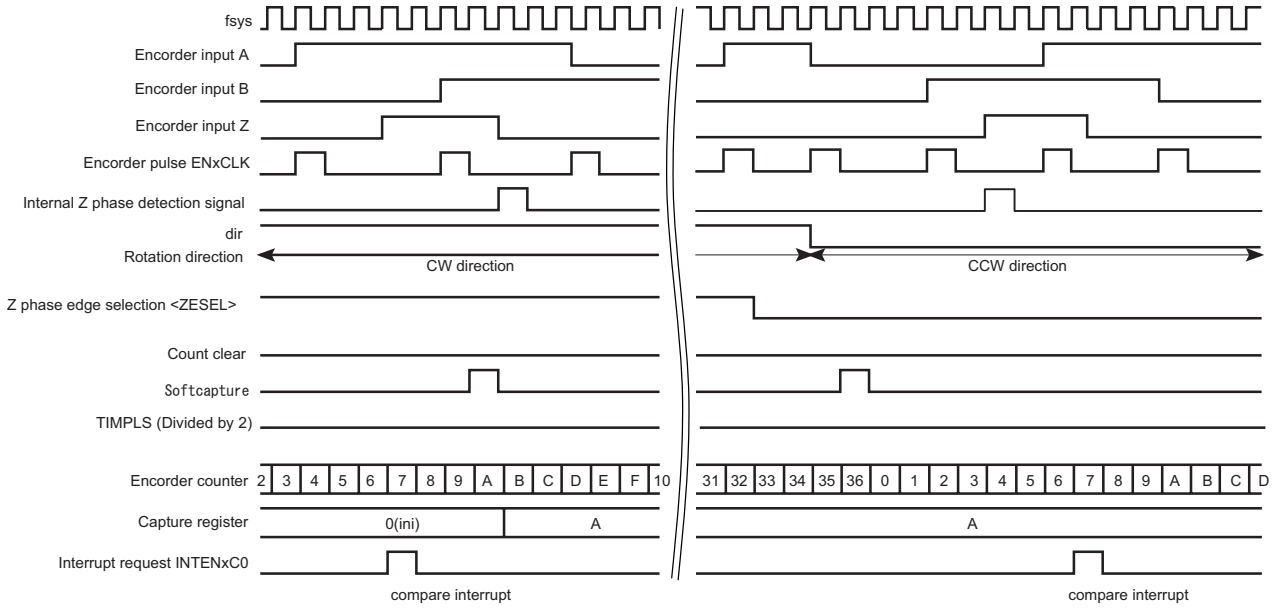
- The encoder counter always counts up; it is cleared to "0" on ENCLK. When the encoder counter has reached to "0xFFFFFF", it wraps around to "0".
- When <ENCLR> is set to 1, the encoder counter is cleared to "0".
- ENCLK captures the value of the encoder counter into the ENCNT register. The captured counter value can be read out of ENCNT.
- Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of <ENINT>.
- When <ENRUN> is set to "0", <UD> is cleared to "0".
- <REVERR> is set to 1 when the rotation direction has changed. This bit is cleared to "0" on a read.
- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

15.6.1.4 Timer mode

1. If <ZEN> = 1 (<ENINT> = 0x0006)



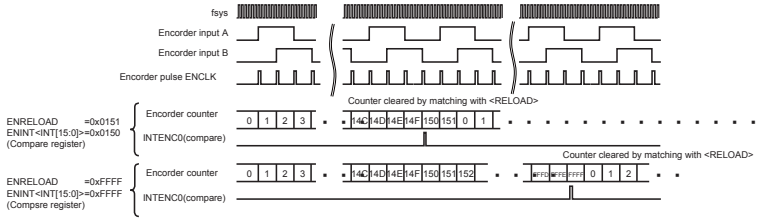
2. If <ZEN> = 0 (<ENINT> = 0x0006)



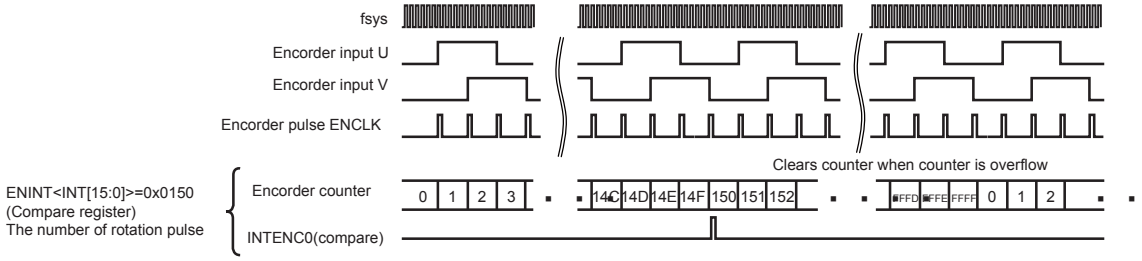
- When <ZEN> = "1", the Z input pin is used as an external trigger. When <ZEN> = "0", no external input is used to trigger the timer.
- The encoder counter always counts up. If <ZEN> = "1", the counter is cleared to "0" on the rising edge of Z when <ZESEL> is set to "0" and a falling edge when <ZESEL> is set to "1". When the encoder counter has reached to "0xFFFFFF", it wraps around to "0".
- When <ENCLR> is set to 1, the encoder counter is cleared to "0".
- Z-Detected causes the value of the encoder counter to be captured into the ENCNT register. The captured counter value can be read out of ENCNT.
- Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of <ENINT>.
- When <ENRUN> is set to "0", <UD> is cleared to "0".
- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

15.6.2 Counter and interrupt generate operation when <CMPEN> = 1

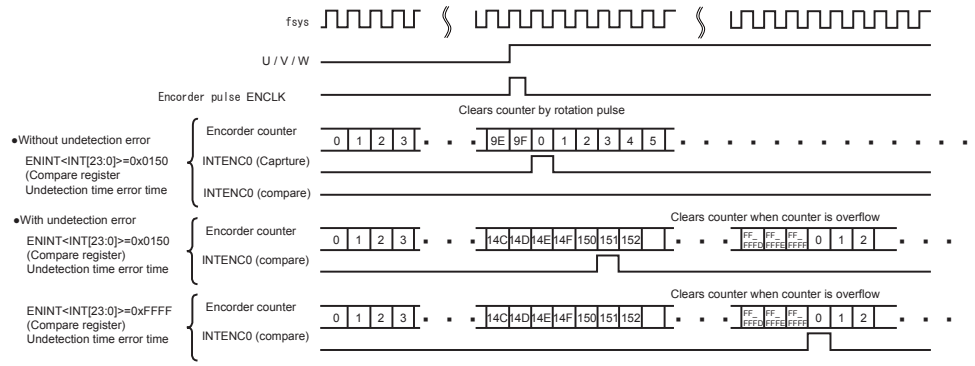
15.6.2.1 Encoder mode



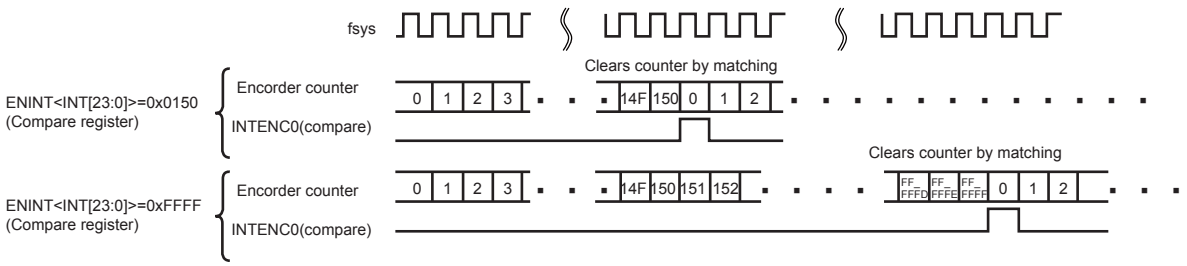
15.6.2.2 Sensor mode (event count)



15.6.2.3 Sensor mode (Timer count)



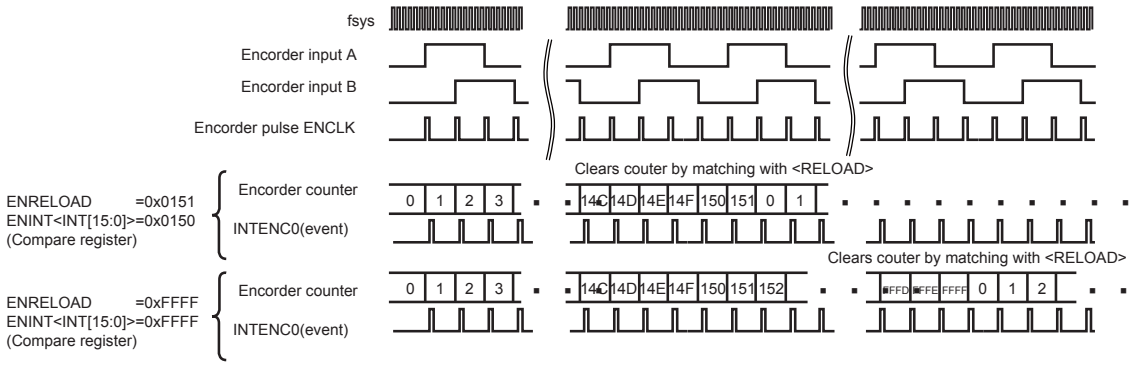
15.6.2.4 Timer mode



15.6.3 Counter and interrupt generate operation when <CMPEN> = 0

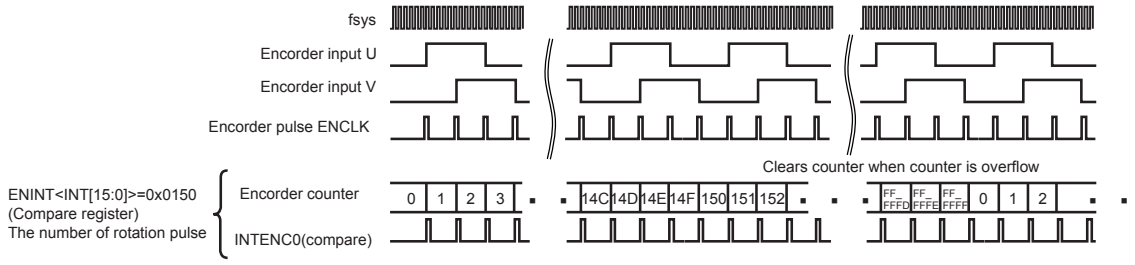
15.6.3.1 Encoder mode

<ENDEV>="000"

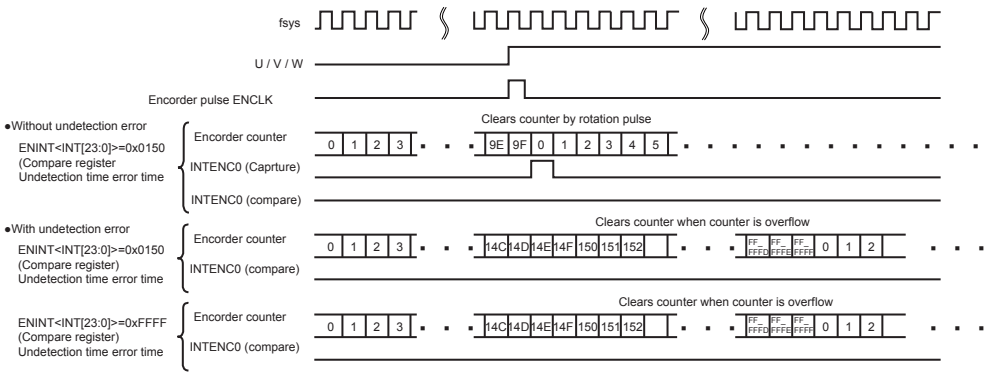


15.6.3.2 Sensor mode (event count)

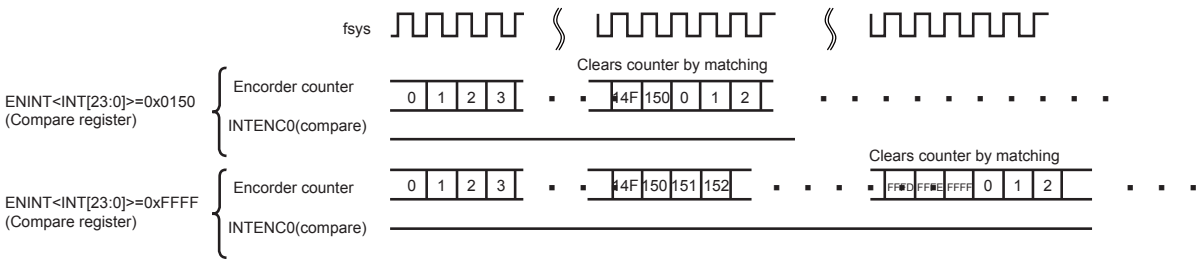
<ENDEV>="000"



15.6.3.3 Sensor mode (Timer count)



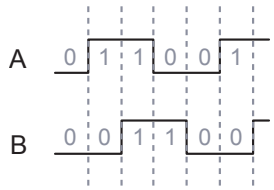
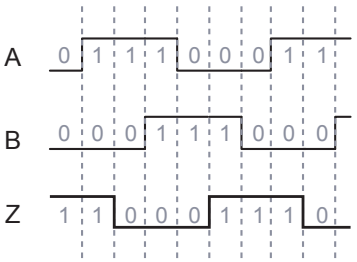
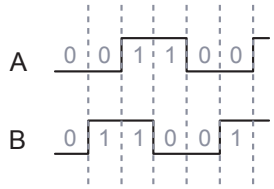
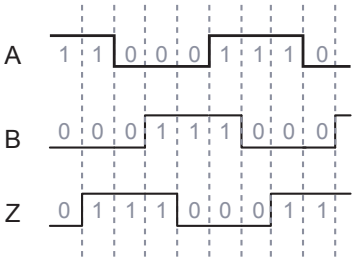
15.6.3.4 Timer mode



15.6.4 Encoder rotation direction

This circuit determines a phase either A-, B- or Z-phase.

It is used as 2-phase input (A,B) and 3-phase input (A,B,Z) in common. When 3-phase input is used, set <P3EN> = "1".

	2-phase input	3-phase input
CW direction		
CCW direction		

15.6.5 Counter Circuit

The counter circuit has a 24-bit up/down counter.

15.6.5.1 Operation Description

Depending on the operation modes, counting, clearing and reloading operation are controlled as described in Table 15-2.

Table 15-2 Counter control

Mode <MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Count	Operation	Counter clear condition	Counter reload condition	Operational range of counter (Reload value)
Encoder mode 00	0	0	A,B	Encoder pulse (ENCLK)	UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD>	-	0x0000 to <RELOAD>
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
	1		A,B,Z		UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD> [3] Z-trigger	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (event count) 01	0	0	U,V		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	0x0000 to 0xFFFF
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
		1	U,V,W		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (Timer count) 10	0	0	U,V	fsys	UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF	-	0x000000 to 0xFFFFFF
		1	U,V,W		UP	[3] Encoder pulse (ENCLK)	-	
Timer mode 11	0	×	-		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <ENINT>	-	0x000000 to 0xFFFFFF
	1		Z		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <ENINT> [4] Z-trigger	-	

Note: The counter value is not cleared by writing "0" to <ENRUN>. If <ENRUN> = "1" is set again, the counter restarts from the counter value which has stopped. If clear the counter value, write "1" to <ENCLR> to execute software clear.

15.6.6 Interrupt

The interrupt consists of four interrupts including Event (divide pulse and capture), Abnormal detecting time, Timer compare and Capture interrupts.

15.6.6.1 Operational Description

When <INTEN> = "1" is set, interrupts occurs by counter value and encoder pulses.

Interrupt factor setting consists of six kinds setting with operation modes and the setting of <CMPEN> and <ZEN>.Table 15-3 shows interrupt factors.

Table 15-3 Interrupt factors

	Interrupt factor	Description	Mode	Interrupt output	Status flag
1	Event count interrupt	When <CMPEN> = 1, the encoder counter counts events (encoder pulses). When it has reached to the value programmed in <ENINT>, an interrupt occurs.	Encoder mode and Sensor mode (event count)	<INTEN> = 1 and <CMPEN> = 1	<CMP>
2	Event interrupt (divide pulse)	An interrupt occurs on each divided clock pulse (1 to 128 divide), which is derived by dividing the encoder pulse by a factor programmed in <ENDEV>.		<INTEN> = 1	Not available
3	Event interrupt (capture interrupt)	An interrupt occurs to indicate that an event (encoder pulse) has occurred, causing the counter value to be captured on the rotation pulse timing.	Sensor mode (Timer count)	<INTEN> = 1	Not available
4	Abnormal detection time error interrupt	When <CMPEN> = 1, the ENC uses a counter that counts up with fsys and is cleared by an event (encoder pulse). If no event occurs for a period of time programmed in <ENINT>, an interrupt occurs.		<INTEN> = 1 and <CMPEN> = 1	<CMP>
5	Timer compare interrupt	When <CMPEN> = 1, an interrupt occurs when the timer has reached to the value programmed in <ENINT>.	Timer mode	<INTEN> = 1 and <CMPEN> = 1	<CMP>
6	Capture interrupt	An interrupt occurs when the counter value has been captured on an external trigger (Z input).		<INTEN> = 1	Not available

In Sensor Timer Count mode and Timer mode, the value of the encoder counter can be captured into the ENCNT register.

The captured counter value can be read out of the ENCNT register.

In Sensor Timer Count mode, the value of the encoder counter is captured into the ENCNT register upon occurrence of an event (encoder pulse). The counter value can also be captured by writing a 1 to <SFTCAP> by software.

In Timer mode, the counter value can be captured by writing a 1 to <SFTCAP> by software. If <ZEN> is set to 1, the counter value can also be captured by an edge of the Z signal input selected according to <ZESEL> by external trigger.

16. Power-on-Reset Circuit (POR)

The power-on-reset circuit (POR) generates a power-on reset signal when power-on.

Power supply voltage is indicated as DVDD5B.

16.1 Structure

Power-on-reset circuit consists of the reference voltage generation circuit, comparators, the VLTD reset circuit and the power-on counter.

This circuit compares a voltage divided by the ladder resistor with a reference voltage generated in the reference voltage generation circuit in the comparator.

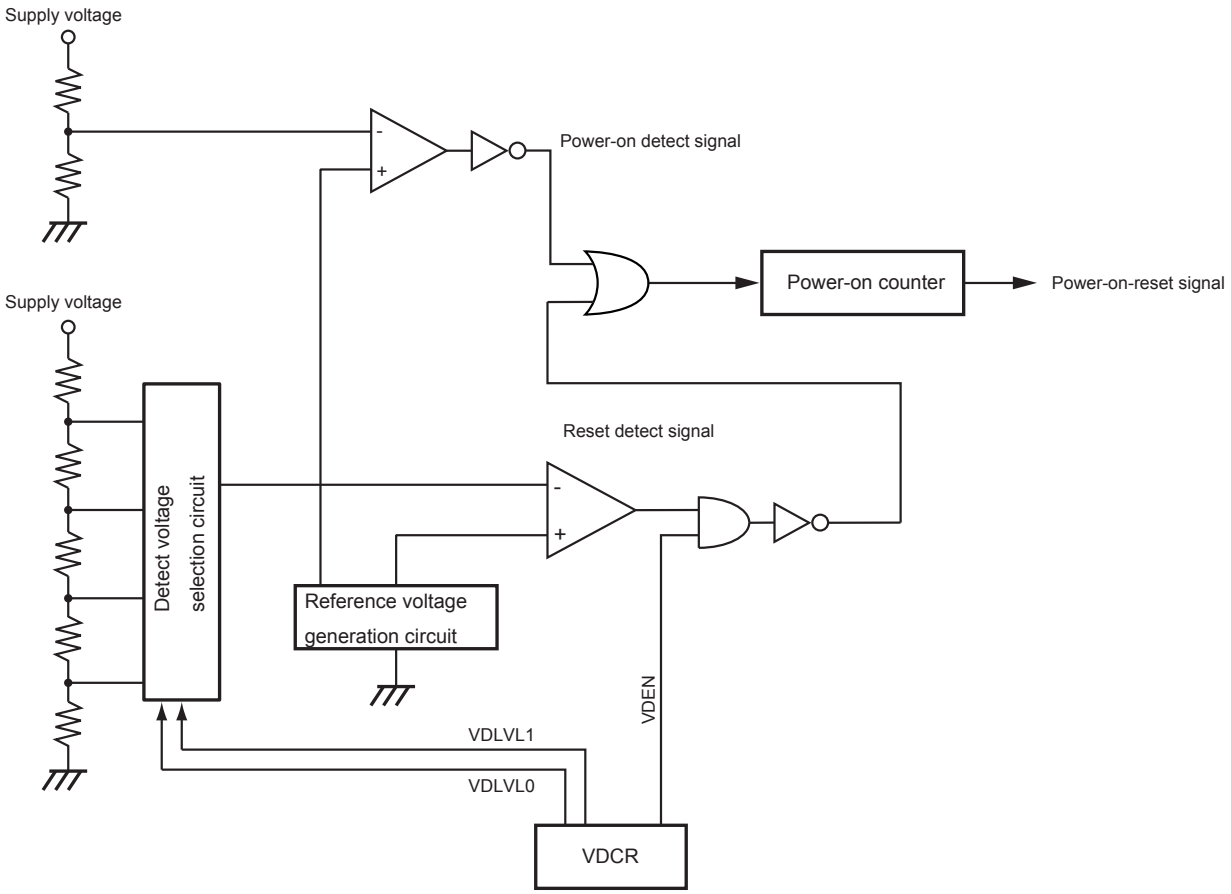


Figure 16-1 Power-on-reset circuit

For details of VDCR in VLTD reset circuit, refer to Section "Voltage detection circuit (VLTD)".

16.2 Function

At power-on, a power-on detection signal generates while power supply voltage is lower than the power-on-reset releasing voltage. Power-on detection signal is released at the timing when DVDD5B is over 3.0 ±0.2 V.

When the power-on detection signal is released and the reset detection signal is also released, the power-on counter starts to operate. After waiting time (approximately 3.2ms ms) has elapsed, the internal reset signal is released.

During the internal reset signal generation, the CPU and the peripheral functions are reset.

When a reset input is not used, supply voltage must be raised to the operational voltage range until the internal reset releasing. If power supply voltage does not reach to the operational voltage range during this period, TMPM3U0FSDMG cannot operate properly.

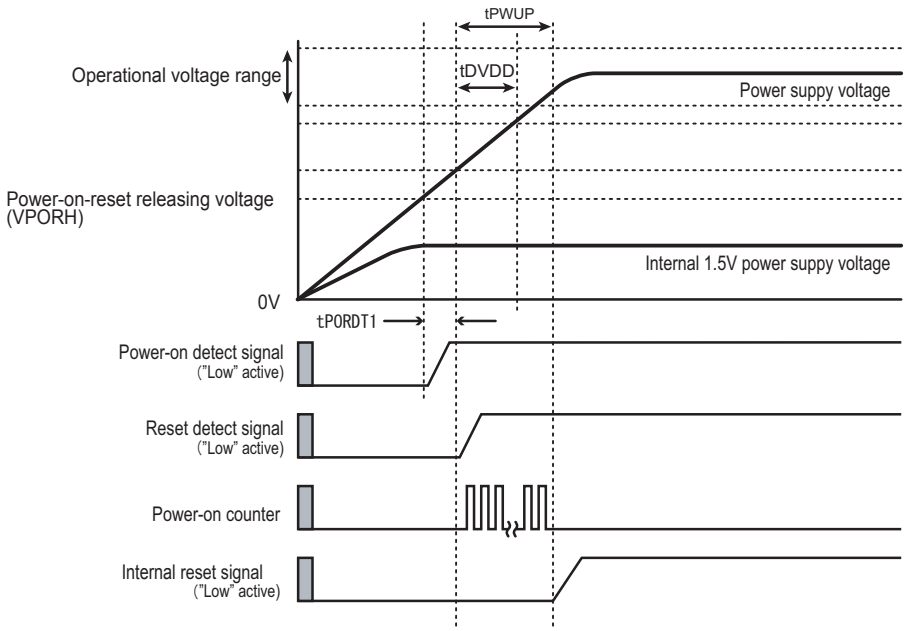


Figure 16-2 Power-on-reset operation timing

Symbol	Parameter	Min	Typ	Max	Unit
tPWUP	Power-on Counter	-	$2^{15}/f_{OSC2}$	-	s
tDVDD	Rising time of power line	-	-	3	ms
VPORH	Power-on Reset releasing voltage	2.8	3	3.2	V
VPORL	Power-on Reset detection voltage	2.6	2.8	3.0	V
tPORDT1	Power-on Reset release sesponse time		30		μS

Note: Since the power-on releasing voltageand the power-on reset detection voltage relatively change,the detection voltage is never reversed.

Note: Due to the fluctuation of supply voltage, the power-on reset circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

17. Low Voltage Detection Circuit (VLTD)

The low voltage detection circuit generates a reset signal by detecting a decreasing voltage.

Note: Due to the fluctuation of supply voltage, the voltage detection circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

17.1 Structure

The low voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (DVDD5B) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDLVL), and the comparator compares it with the reference voltage. When the supply voltage (DVDD5B) becomes lower than the detection voltage (VDLVL), a voltage detection reset signal is generated.

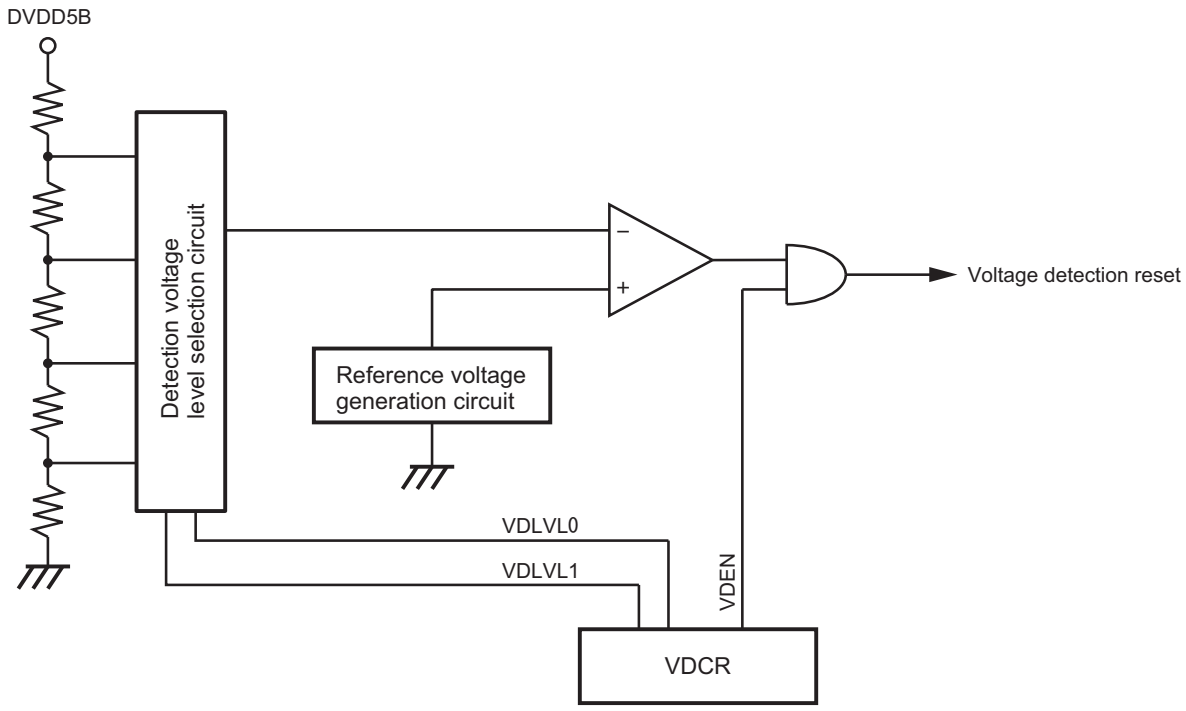


Figure 17-1 Voltage Detection Circuit)

17.2 Registers

17.2.1 Register List

Base Address = 0x4004_0900

Register name		Address(Base+)
Voltage detection control register	VDCR	0x0000

17.2.2 VDCR (Voltage detection control register)

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

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2-1	VDLVL[1:0]	R/W	Selection for detection voltage 00: Reserved 01: 4.1 ± 0.2V 10: 4.4 ± 0.2V 11: 4.6 ± 0.2V
0	VDEN	R/W	Low voltage detection operation 0: Disabled 1: Enabled

Note: VDCR is initialized by a power-on reset or an external reset input.

17.3 Operation Description

17.3.1 Control

The voltage detection circuit is controlled by voltage detection control registers.

17.3.2 Function

The detection voltage can be selected by VDCR<VDLVL[1:0]>. Enabling/disabling the voltage detection can be programmed by VDCR<VDEN>. After the voltage detection operation is enabled, When the supply voltage (DVDD5B) becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection reset signal is generated.

17.3.2.1 Enabling/disabling the voltage detection operation

Setting VDCR<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation. VDCR<VDEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (DVDD5B) is lower than the detection voltage VDCR<VDLVL[1:0]>, setting VDCR<VDEN> to "1" generates reset signal at the time.

17.3.2.2 Selecting the detection voltage level

Select a detection voltage at VDCR<VDLVL[1:0]>.

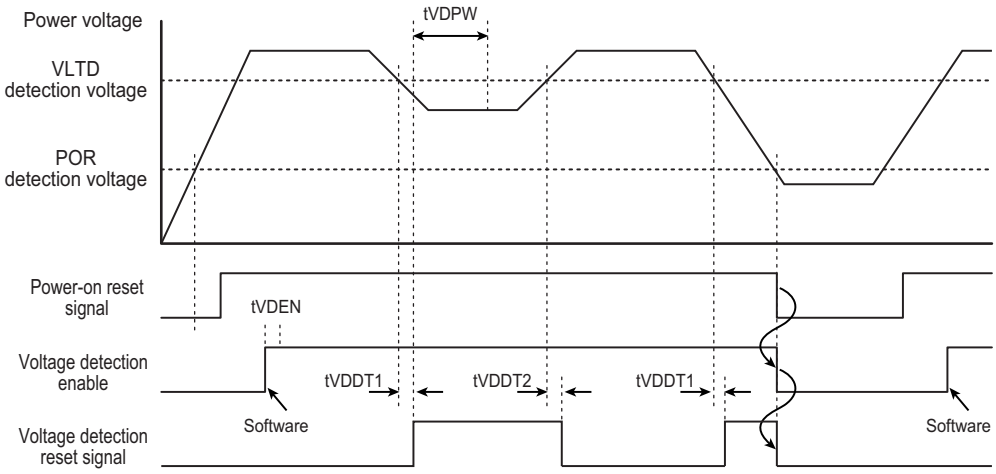


Figure 17-2 Voltage Detection Timing

Symbol	Parameter	Min	Typ	Max	Unit
tVDEN	Setup time after enabling voltage detection	-	40	-	μs
tVDDT1	Voltage detection response time	-	40	-	
tVDDT2	Voltage detection releasing time‘	-	40	-	
tVDPW	Voltage detection minimum pulse width	45	-	-	

18. Ocsillation Frequency Detector (OFD)

The oscillation frequency detector generates a reset for I/O if the oscillation of external high frequency for CPU clock exceeds the detection frequency range.

The oscillation frequency detection is controlled by OFDCR1, OFDCR2 registers and the detection frequency range is specified by OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON which are the detection frequency setting registers. The lower detection frequency is specified by OFDMNPLLOFF/OFDMNPLLON registers and the higher detection frequency is specified by OFDMXPLLOFF/OFDMXPLLON registers.

When the oscillation frequency detection is enabled, writing to OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers is disabled. Therefore, the setting the detection frequency to these registers should be done when the oscillation frequency detection is disabled. And writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers is controlled by OFDCR1 register. To write OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers, the write enable code "0xF9" should be set to OFDCR1 beforehand. To enable the oscillation frequency detector, set "0xE4" to OFDCR2 after setting "0xF9" to OFDCR1. Since the oscillation frequency detection is disabled after an external re-set input, power on reset or VLTD reset, write "0xF9" to OFDCR1 and write "0xE4" to OFDCR2 register to ena-ble its function.

When the TMPM3U0FSDMG detects the out of frequency by lower and higher detection frequency setting registers, all I/Os become high impedance by reset. In case of PLLOFF, OFDMNPLLOFF and OFDMXPLLOFF registers are valid for detection and the setting value of OFDMNPLLON/OFDMXPLLON registers are ignored. In case of PLLON, OFDMNPLLON and OFDMXPLLON registers are valid for detection and the setting value of OFDMNPLLOFF/OFDMXPLLOFF registers are ignored. By the oscillation frequency detection reset, all I/Os except power supply pins, $\overline{\text{RESET}}$ pin and MODE pin become high impedance and the CPU is also initialized. As the CG registers are initialized, so the PLL operation of the system clock is disabled and the clock operation is restarted after switching an oscillator to on chip oscillation (fosc2).

Since all registers for oscillation frequency detector (OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON) are not initialized by the reset generated from oscillation frequency detector.

And so when the oscillation frequency detection reset is generated, the system clock is switched to on chip oscillation and the reset sequence is executed just as the oscillation frequency detection is enabled.

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

Note: The oscillation frequency detection reset is available only in NORMAL and IDLE modes. In STOP mode, the oscillation frequency detection reset is disabled automatically.

Note: When the PLL is controlled (enabled or disabled) by the CGPLLSEL register or when the system clock is changed (fosc1 or fosc2) by the <OSCSEL> of CGOSCCR register, the OFD must be disabled beforehand. If OFD reset is generated with PLL-ON, the detection frequency setting registers (OFDMNPLLON/OFDMXPLLON) are automatically switched over to OFDMNPLLOFF/OFDMXPLLOFF.

18.1 Block diagram

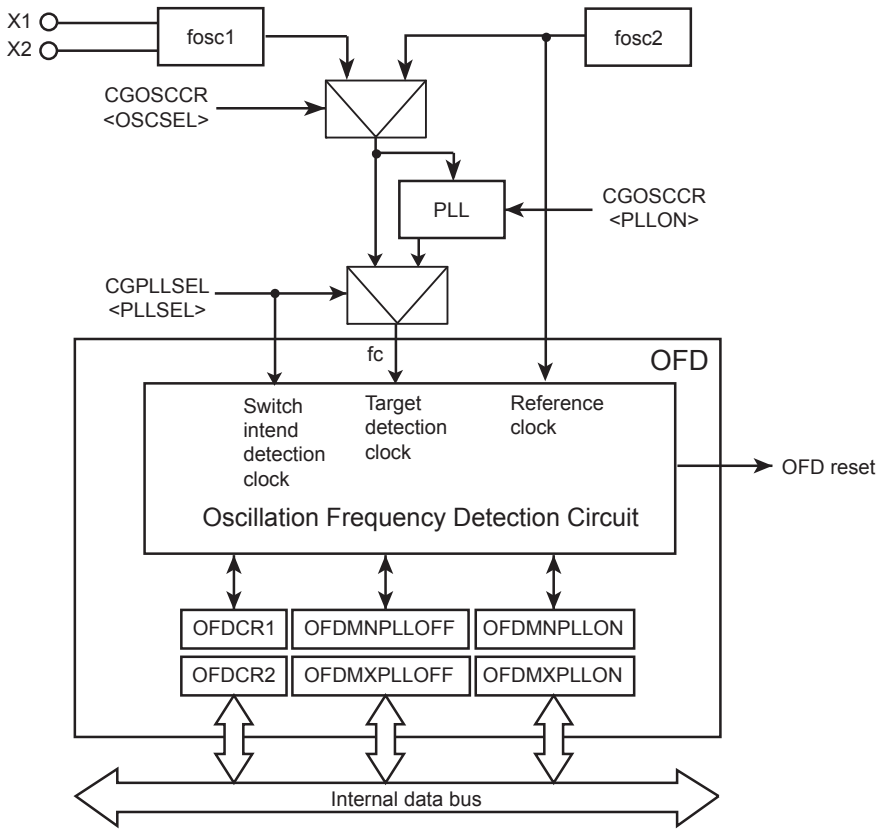


Figure 18-1 Ocsillation Frequency Detector Block diagram

18.2 Registers

18.2.1 Register List

Base Address = 0x4004_0800

Register name		Address(Base+)
Oscillation frequency detection control register 1	OFDCR1	0x0000
Oscillation frequency detection control register 2	OFDCR2	0x0004
Lower detection frequency setting register (PLL OFF)	OFDMNPLLOFF	0x0008
Lower detection frequency setting register (PLL ON)	OFDMNPLLON	0x000C
Higher detection frequency setting register (PLL OFF)	OFDMXPLLOFF	0x0010
Higher detection frequency setting register (PLL ON)	OFDMXPLLON	0x0014

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

18.2.1.1 OFDCR1 (Oscillation frequency detection control register)

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

Note: OFDCR1 is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

18.2.1.2 OFDCR2 (Oscillation frequency detection 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.

Note: OFDCR2 is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

18.2.1.3 OFDMNPLLOFF (Lower detection frequency setting register (In case of PLL OFF))

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMNPLL-OFF[8:0]	R/W	Sets internal lower detection frequency. After reset value which is set for the reference clock is 9.7MHz ± 5% and the target detection clock is 10MHz ± 5%.

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

Note: OFDMNPLLOFF is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

18.2.1.4 OFDMNPLLON (Lower detection frequency setting register (In case of PLL ON))

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMNPLLON [8:0]	R/W	Sets external lower detection frequency. After reset value which is set for the reference clock is 9.7MHz ± 5% and the target detection clock is 40MHz ± 5%.

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

Note: OFDMNPLLON is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

18.2.1.5 OFDMXPLLOFF (Higher detection frequency setting register (In case of PLL OFF))

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMXPLL-OFF[8:0]	R/W	Sets internal higher detection frequency. After reset value which is set for the reference clock is 9.7MHz ± 5% and the target detection clock is 10MHz ± 5%.

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

Note: OFDMXPLLOFF is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

18.2.1.6 OFDMXPLLON (Higher detection frequency setting register (In case of PLL ON))

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMXPLLON [8:0]	R/W	Sets external higher detection frequency. After reset value which is set for the reference clock is 9.7MHz ± 5% and the target detection clock is 41MHz.

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

Note: OFDMXPLLON is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

18.3 Operational Description

18.3.1 Setting

Registers of OFD are initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset. All register except OFDCR1 can not be written by reset. They are able to be written by writing "0xF9" to OFDCR1.

The range of detection frequency is setting by OFDMNPLLON/OFDMXPLLON or OFDMNPLLOFF/ OFDMXPLLOFF for each target clock. Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection.

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

OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of CGPLLSEL<PLLSEL>.

When STOP mode is executed with OFDCR2=0xE4, the oscillation frequency detection is automatically disabled. After releasing STOP mode and warming up period, the oscillation frequency detection is enabled. The oscillation frequency detection is available only in NORMAL and IDLE mode. Table 18-1 shows the availability of oscillation frequency detector.

Table 18-1 Availability of oscillation frequency detector

Operating Mode	Oscillation Frequency Detection (OFDCR2=0xE4)	All I/Os condition after Oscillation Frequency Detection RESET (Except power supply, $\overline{\text{RESET}}$, MODE, pins)
NORMAL	Available	High impedance
IDLE	Available	High impedance
STOP (Including warming up period)	Oscillation Frequency Detection is disabled automatically.	
Reset by oscillation frequency detection reset	Available	High impedance
Watchdog timer reset SYSRESETREQ reset	Available	High impedance
RESET by external reset power on reset VLTD reset	Disable	-

18.3.2 Operation

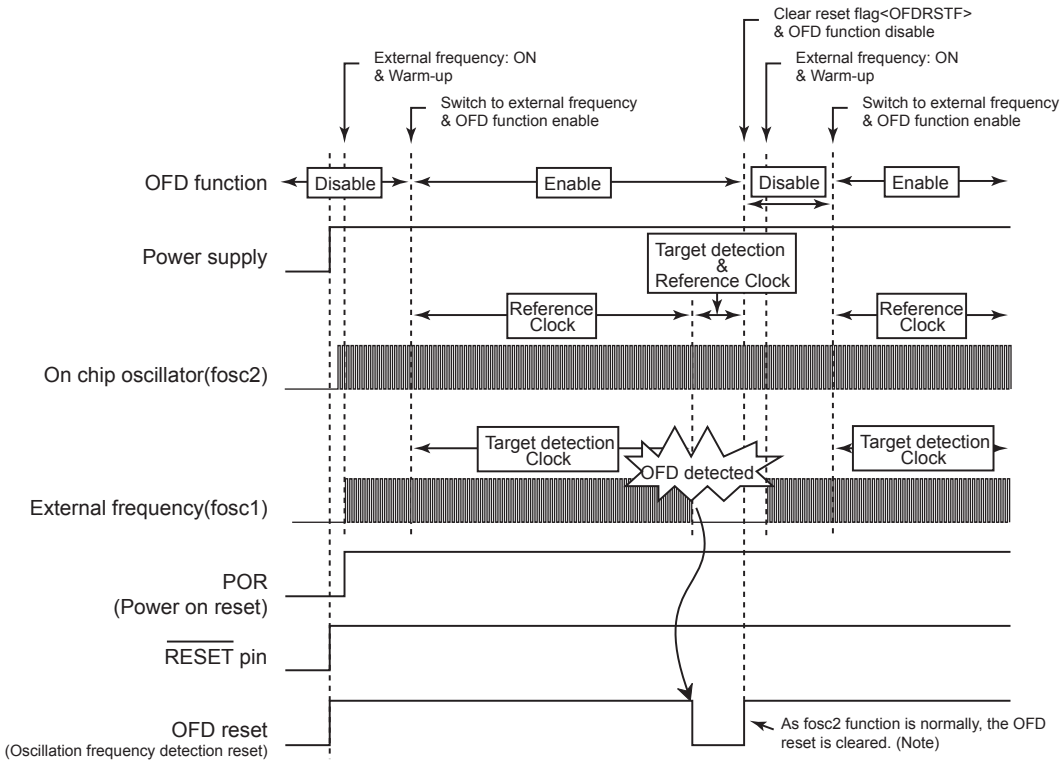
From the operation start-up to detection start-up, time length as two cycle of detecting clock is needed. And the Detecting cycle is 128/reference clock frequency.

When generating reset is enabled, OFD generates reset if the target clock frequency exceeds the frequency limit set by OFDMNPLLON/OFDMXPLLON and OFDMNPLLOFF/OFDMXPLLOFF. From detection of abnormal to reset generation, time length as one cycle of detecting clock is needed. The reset generated by OFD does not make itself and OFD continues detected operation.

Therefore, fosc is initialized to fosc2 and the target clock(fc) changes to fosc2 and detected operation is continued in PLL OFF.

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

Note: The target clock is set without 10MHz into OFDMNPLLOFF and OFDMXPLLOFF. (eg, setting to 8MHz value to these registers) And when the target clock frequency is fosc1 and the reset generated by OFD, the reset might be generated continuously due to as abnormal detection until OFD detects correctly.



Note: The target clock is set without 10MHz into OFDMNPLLOFF and OFDMXPLLOFF. (eg, setting to 8MHz value to these registers) And when the target clock frequency is fosc1 and the reset generated by OFD, the reset might be generated continuously due to as abnormal detection until OFD detects correctly.

Figure 18-2 Example of oscillation frequency detection operation

20.3.3 Detection Frequency

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

The upper and lower limit of detecting frequency are calcurated by the maxmum error of a target clock and a reference clock.

By the way of rounding the calcurated result when OFDMNPLLON/OFDMNPLLOFF and OFDMXPLLON/OFDMXPLLOFF are decided, the upper and lower limit of detecting and undetecting range shown as follows. The way of rouding is selected depending on the uneveness of the detected clock.

- In case of rounding up OFDMXPLLON/OFDMXPLLOFF and rouding down OFDMNPLLON/OFDMNPLLOFF

  The target clock is higher than the upper limit of undetecting ragne and lower than the lower limit of undetecting range.
- In case of rounding down OFDMXPLLON/OFDMXPLLOFF and rouding down OFDMNPLLON/OFDMNPLLOFF

  The target clock is lower than the upper limit of undetecting ragne and higher than the lower limit of undetecting range.

How to calculate the setup value of OFDMXPLLOFF/OFDMNPLLOFF is shown below when the target clock error is ±5% (undetecting range) and the reference clock error is ± 5%. In this example, OFDMXPLLOFF is rounded up and OFDMNPLLOFF is rounded down.(From "a" to "h" corresponds to "Figure 18-3 Example of detection frequency range (in case of 10MHz)")

target clock	10MHz ± 5%	Max.10.5MHz	-------- c
		Min. 9.5MHz	-------- b
reference clock	9.7MHz ± 5%	Max.10.185MHz	-------- f
		Min. 9.215MHz	-------- e

OFDMXPLLOFF = c ÷ e × 32 = 36.46... = 37 (Rounding up to nearest decimal) = 0x25

OFDMNPLLOFF = b ÷ f × 32 = 29.85... = 29 (Rounding down to nearest decimal) = 0x1D

At this time, the detecting range is calculated shown below.

a = e × OFDMNPLLOFF ÷ 32 = 8.35

d = f × OFDMXPLLOFF ÷ 32 = 11.78

And the undetecting range is calculated shown below.

g = e × OFDMXPLLOFF ÷ 32 = 10.65

h = f × OFDMNPLLOFF ÷ 32 = 9.23

Setting "0x25" to the register OFDMXPLLOFF and "0x1D" to the register OFDMNPLLOFF, when the target clock of higher than 11.78MHz or lower than 8.35MHz is detected, the oscillation frequency detector outputs a reset signal. And when the target clock of higher than 9.23MHz and lower than 10.65MHz is detected, the oscillation frequency detector does not output a reset signal.

Figure 18-3 shows the detection or undetectable and detectable frequency range.

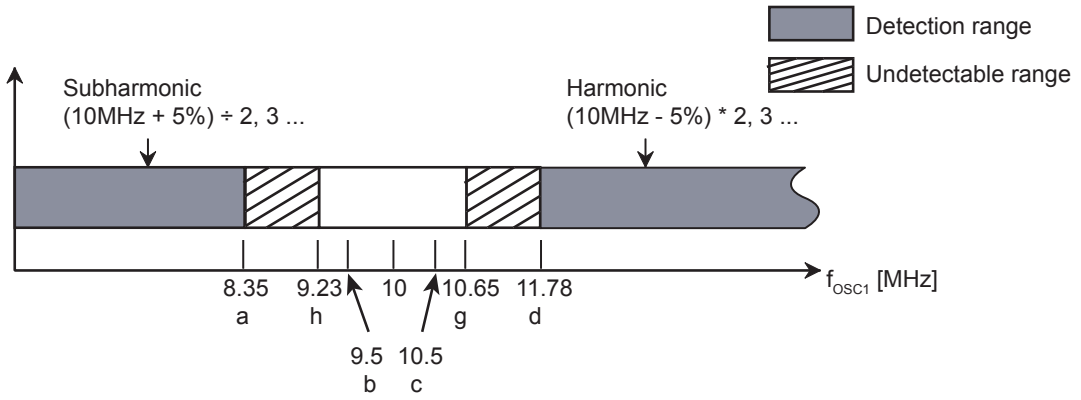


Figure 18-3 Example of detection frequency range (in case of 10MHz)

18.3.4 Available Operation Mode

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

18.3.5 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.

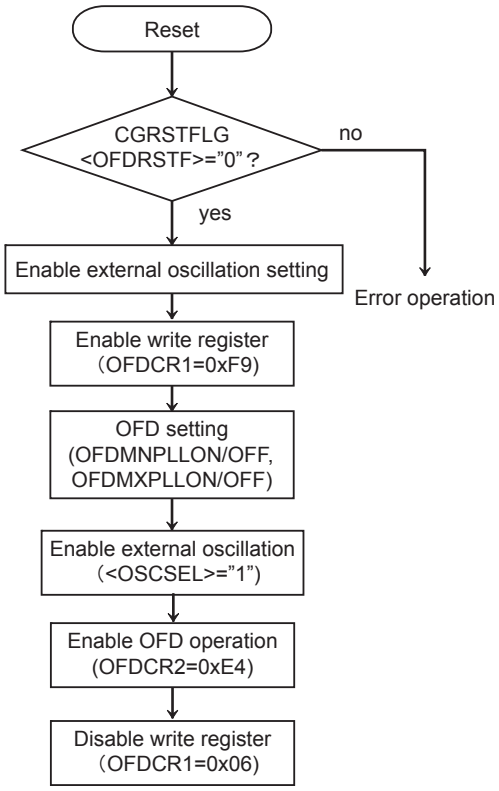


Figure 18-4 Example of operational procedure

19. 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 ($\overline{\text{WDTOUT}}$).

19.1 Configuration

Figure 21-1shows the block diagram of the watchdog timer.

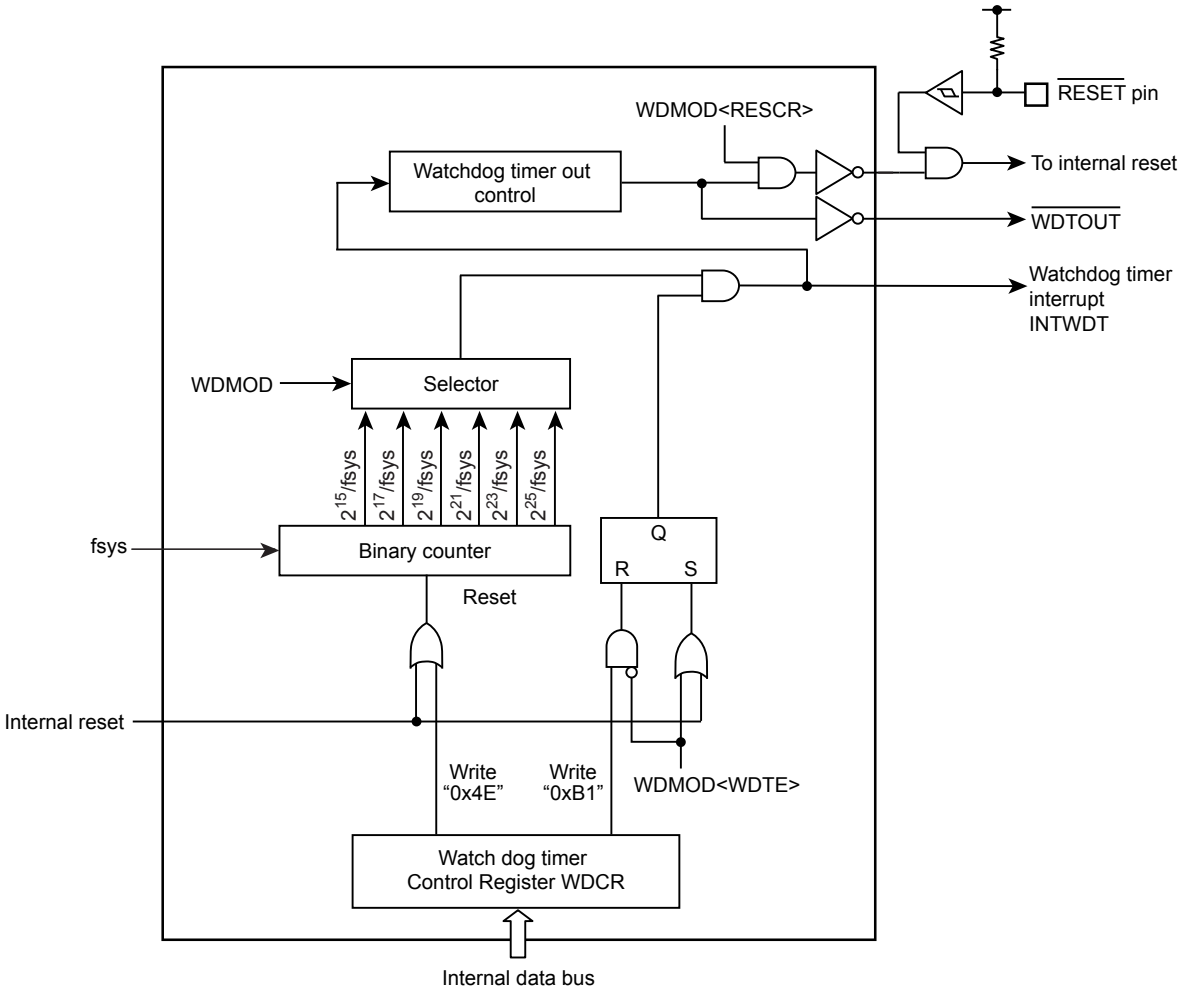


Figure 19-1 Block Diagram of the Watchdog Timer

19.2 Register

The followings are the watchdog timer control registers and addresses.

Base Address = 0x4004_0000

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

19.2.1 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 to Table 19-1) 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 19-1 Detection time of watchdog timer (fc = 40MHz)

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

19.2.2 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

19.3 Operations

19.3.1 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 ($\overline{\text{WDTOUT}}$).

19.3.2 Operation Mode and Status

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

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

The watchdog timer cannot be used as the high-speed frequency clock is stopped. Before transition to below modes, the watchdog timer should be disabled.In IDLE mode, its operation depends on the WDMOD <I2WDT> setting.

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

19.4 Operation when malfunction (runaway) is detected

19.4.1 INTWDT interrupt generation

In the Figure 19-2 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($\overline{\text{WDTOUT}}$).

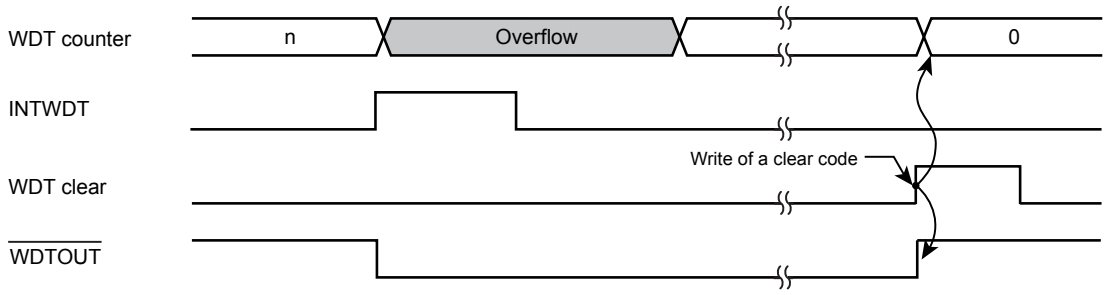


Figure 19-2 INTWDT interrupt generation

19.4.2 Internal reset generation

Figure 19-3 shows the internal reset generation (WDMOD<RESCR>="1").

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

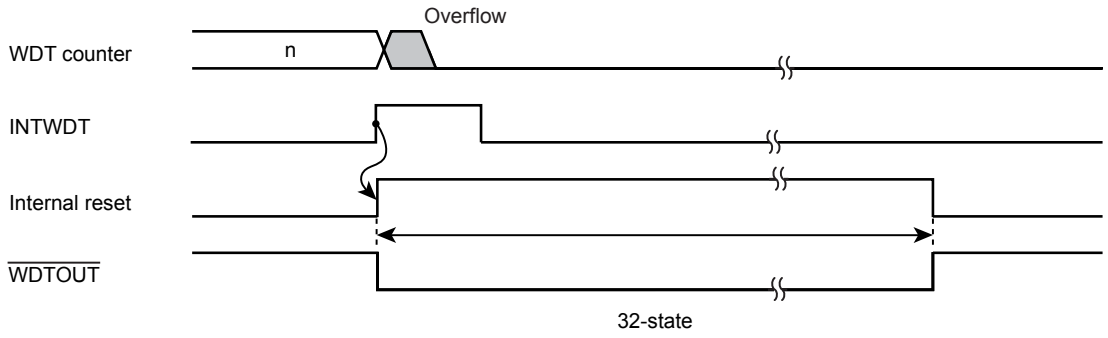


Figure 19-3 Internal reset generation

19.5 Control register

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

19.5.1 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.

19.5.2 Watchdog Timer Control Register(WDCR)

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

19.5.3 Setting example

19.5.3.1 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).

19.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

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

19.5.3.3 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).

19.5.3.4 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	−	−	−	−

20. Flash Memory Operation

This section describes the hardware configuration and operation of Flash memory. In this section, "1-word" means 32 bits.

20.1 Features

20.1.1 Memory Size and Configuration

Table 20-1 and Figure 20-1 show a built-in memory size and configuration of TMPM3U0FSDMG.

Table 20-1 Memory size and configuration

Memory size	Block configuration				# of words per page	# of pages	Write time		Erase time	
	128 KB	64 KB	32 KB	16 KB			1 page	Total area	Block erase	Chip erase
64 KB	-	-	2	-	32	512	1.25ms	0.64 sec	0.1sec	0.2 sec

Note: The above values are theoretical values not including data transfer time. The write time per chip depends on the write method used by a user.

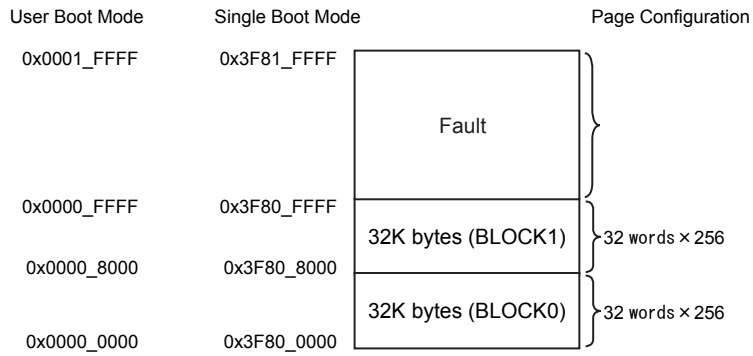


Figure 20-1 Block configuration

Flash memory configuration units ares described as "block" and "page".

- Page

  One page is 32 words. Same address [31:7] is used in a page. First address of the group is [6:0] = 0 and the last address of the group is [6:0] = 0x7F.

- Block

  One block is 32KB and flash memory is consists of two blocks.

Write operation is performed per page. The write time per page is 1.25ms. (Typ.)

Erase is performed per block (auto block erase command use) or performed on entire flash memory (use of auto chip erase command). Erase time varies on commands. If auto block command is used, the erase time will be 0.1 sec per block (Typ.). If the auto chip erase command is used to erase entire area, the time will be 0.2 sec (Typ.).

In addition, the protect function can be used per block. For detail of the protect function, refer to "20.1.5 Protect/Security Function".

20.1.2 Function

Flash memory built-in this device is generally compliant with the JEDEC standards except for some specific functions. Therefore, if a user is currently using a flash memory as an external memory, it is easy to implement the functions into this device. Furthermore, to provide easy write or erase operation, this product contains a dedicated circuit to perform write or chip erase automatically.

JEDEC compliant functions	Modified, added, or deleted functions
・Automatic programming ・Automatic chip erase ・Automatic block erase ・Data polling/toggle bit	<Modified> Block write/erase protect (only software protection is supported) <Deleted> Erase resume - suspend function

20.1.3 Operation Mode

20.1.3.1 Mode Description

This device provides the single chip mode and single boot mode. The single chip mode contains the normal mode and user boot mode. Figure 20-2 shows the mode transition.

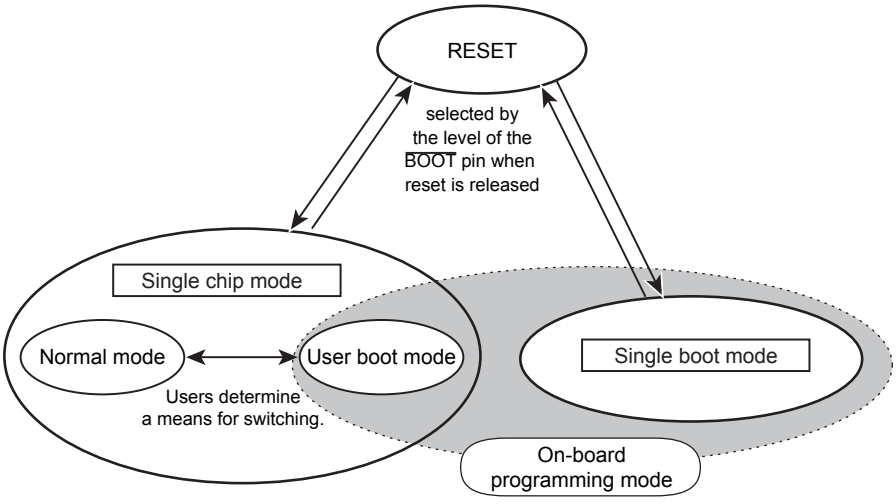


Figure 20-2 Mode transition

(1) Single chip mode

The single chip mode is a mode where the device can boot-up from Flash memory after reset. The mode contains two sub-modes in below.

- Normal mode

  The mode where user application program is executed.
- User boot mode

  The mode where flash memory is re-programmed on the user's set.

Users can switch the normal mode to user boot mode freely. For example, a user can set if PA0 of port A is "1", the mode is the normal mode. If PA0 of port A is "0", the mode is the user boot mode. The user must prepare a routine program in the application program to determine the switching.

(2) Single boot mode

The mode where flash memory can boot-up from the built-in BOOT ROM (Mask ROM) after reset.

The BOOT ROM contains the algorithm that can rewrite Flash memory via serial port of this device on the user's set. With connecting the serial port to external host, data transfer is performed in above-mentioned protocol and re-programed Flash memory.

(3) On-board programming mode

The user boot mode and single boot mode are the modes where flash memory can be re-programmable on the user's set. These two modes are called "on-board programming mode".

20.1.3.2 Mode Determination

Either the single chip or single boot operation mode can be selected by the level of the $\overline{\text{BOOT}}$ pin when reset is released.

Table 20-2 Operation mode setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

20.1.4 Memory Map

Figure 20-3shows a comparison of the memory map in the single chip mode and single boot mode. In the single boot mode, built-in Flash memory is mapped to 0x3F80_0000 and subsequent addresses, and the built-in BOOT ROM is mapped to 0x0000_0000 through 0x0000_0FFF.

Flash memory and RAM addresses are shown below.

FLASH size	RAM size	FLASH address	RAM address
64KB	4KB	0x0000_0000 to 0x0000_FFFF(single chip mode) 0x3F80_0000 to 0x3F80_FFFF(single boot mode)	0x2000_0000 to 0x2000_0FFF

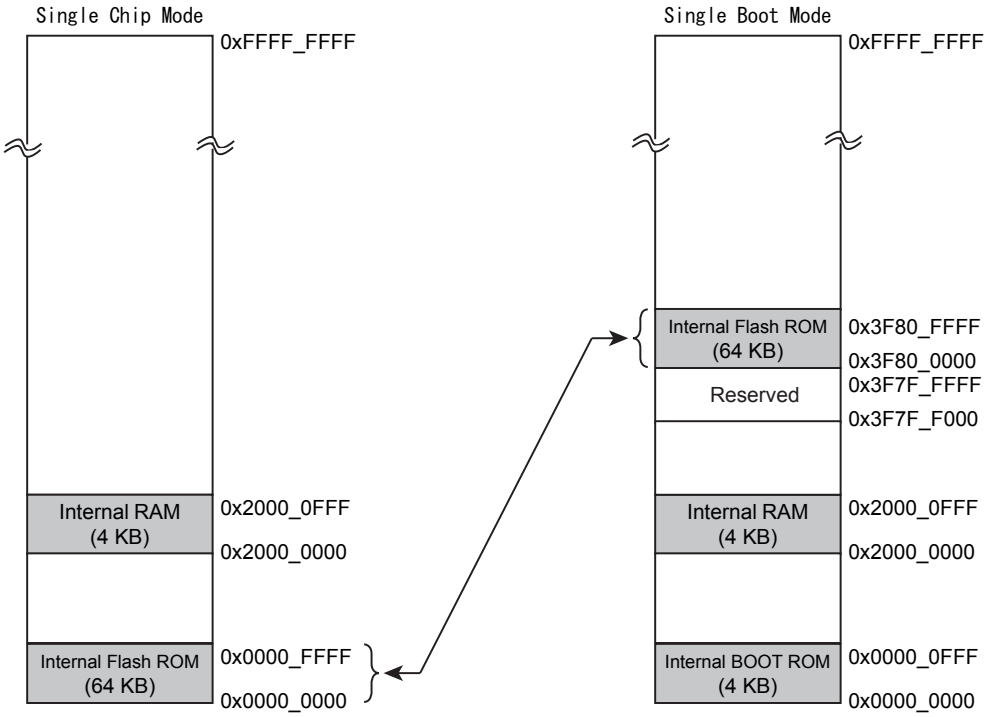


Figure 20-3 Comparison of memory map

20.1.5 Protect/Security Function

This device has the protect and security functions for Flash memory.

1. Protect function

   The write/erase operation can be inhibited per block.

2. Security function

   The read operation from a flash writer can be inhibited.

   Usage restrictions on debug functions

20.1.5.1 Protect Function

This function inhibits the write/erase operation per block.

To enable the protect function, a protect bit corresponding to a block is set to "1" using the protect bit program command. If a protect bit is set to "0" using the protect bit erase command, a block protect can be can-celed. The protect bit can be monitored with FCPSRA<BLK1><BLK0>.

A program of protect bit can be programmed by 1-bit unit and can be erased by 4-bit unit. For detail of programming/erasing of protect bits, refer to "20.2.5 Command Description".

20.1.5.2 Security Function

Table 20-3 shows operations when the security function is enabled.

Table 20-3 Operations when the security function is enabled.

Item	Description
Read flash memory	CPU can read flash memory.
Debug port	JTAG, serial wire or trace communication is disabled.
Command execution to Flash memory	Command write to flash memory is not accepted. If a user tries to erase a protect bit, chip erase is executed and all protect bits are erased.

The security function is enabled under the following conditions;

1. FCSECBIT<SECBIT> is set to "1".
2. All protect bits (FCPSRA<BLK>) are set to "1".

FCSECBIT<SECBIT> is set to "1" by the Cold Reset(Power OnReset) . Rewriting of FCSECBIT <SECBIT> is described in below.

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCSECBIT
2. Write data within 16 clocks after the operation of item 1.

20.1.6 Register

20.1.6.1 Register List

Base Address = 0x41FF_F000

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0010
Flash Interface controlregister	FCCR	0x001C
Flash status register	FCSR	0x0020
Flash protect status register A	FCPSRA	0x0030

20.1.6.2 FCCR (Flash Interface control register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	FLBOFF	R/W	Control of Flash Interface with instruction Buffer (Note) 0: Enable Instruction Buffer 1: Disable Instruction Buffer (with Buffer clear) This bit is a functional bit for controlling the Flash Interface . To use Intstuction Buffer, set "0". To not use Instruction Buffer ,set to "1". In TMPM3U0FSDMG, it must be set "0" for Flash accessing.

Note : In TMPM3U0FSDMG, after Flash programing or Flash Erasing ,it should be Clearing Instruction buffer by this functional bit or insert a reset signal .

Instruction Buffer Clearing operation as following.

after excuting FCCR<FLBOFF>="1" , set FCCR<FLBOFF>="0" again on RAM.

20.1.6.3 FCSR (Flash status register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	RDY/BSY	R	Ready/Busy (Note) 0: Busy (during auto operation) 1: Ready (auto operation ends) This bit is a function bit to monitor flash memory from CPU. While flash memory is in auto operation, this bit outputs "0" to indicate that flash memory is busy. Once auto operation is finished, this bit becomes ready state and outputs "1". Then next command is accepted. If a result of auto operation is failed, this bit outputs "0" continuously. The bit returns to "1" by hardware reset.

Note: Make sure that flash memory is ready before commands are issued. If a command is issued during busy, not only the command is not sent but also subsequent commands may not be accepted. In that case, use hardware reset to return. Hardware reset needs 0.5 µs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset.

20.1.6.4 FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SECBIT	R/W	Security bit 0: Security function setting is disabled. 1: Security function setting is enabled.

Note: This register is initialized by Cold Reset(Power OnReset) .

20.1.6.5 FCPSRA (Flash protect status register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	BLK1	BLK0
After reset	0	0	0	0	0	0	(Note)	(Note)

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1-0	BLK1- BLK0	R	Protection status of Block1 to 0 0: Not protected 1: Protected Protect bit values correspond to protect status of each block. If corresponding bit indicates "1", corresponding block is in the protection status. A block in the protection status cannot be re-programmable.

Note : A value will correspond to the protection status.

20.2 Detail of Flash Memory

In on-board programming, the CPU executes commands for reprogramming or erasing Flash memory. This reprogramming/erase control program should be prepared by the user beforehand. Since Flash memory content cannot be read while Flash memory is being written or erased, it is necessary to run the reprogram/erase control program on the built-in RAM. Do not generate interrupt/fault except reset to avoid abnormal program termination.

20.2.1 Function

Flash memory is generally compliant with the JEDEC standards except for some specific functions. However; a method of address designation of operation command is different from standard commands.

If write/erase operation is executed, commands are input to flash memory using 32-bit (1-word) store instruction command. After command input, write or erase operation is automatically executed in inside.

Table 20-4 Flash memory function

Main function	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of Flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Write/erase protect	The write or erase operation can be individually inhibited for each block.

Note: In TMPM3U0FSDMG, after Flash programing or Flash Erasing ,it should be Clearing Instruction buffer. Pls refter "22.1.6.2 FCCR (Flash Interface control register)" for a clear method.

Note: Check the FCSR<RDY/BSY> to make sure each command sequence end such as Flash writing,Flash Erase, Protection bit program, Protection bit Erase . and then hold for 200 μs or more before reading data from Flash memory or starting instruction fetch.

20.2.2 Operation Mode of Flash Memory

Flash memory provides mainly two types of operation modes;

- The mode to read memory data (Read mode)
- The mode to erase or rewrite memory data automatically (Automatic operation mode)

After power-on, after rest or after automatic operation mode is finished normally, Flash memory becomes read mode. Instruction stored in Flash memory or data read is executed in the read mode.

If commands is input during the read mode, the operation mode becomes the automatic operation. If the command process is normally finished, the operation mode returns to the read mode except the ID-Read command. During the automatic operation, data read and instruction execution stored in Flash memory cannot be performed.

If command process is abnormally finished then the operation mode should forcibly return to read mode. In this case, use the read command, read/reset command or hardware reset.

20.2.3 Hardware Reset

A hardware reset means a Cold Reset(Power OnReset) or warm reset to use returning to the read mode when the automatic programming/erase operation is focibly cancelled, or automatic operation abnormally ends.

If the hardware reset occurs during the automatic operation, Flash memory stops the automatic operation and returns to the read mode. If a hardware reset is generated during Flash memory automatic program/erase operation, the hardware reset needs 0.5 μs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset . Note that if a hardware reset occurs during the automatic operation, data write operation is not executed properly. Set write operation again.

For detail of the reset operation, refer to "Reset". After a given reset input, CPU will read the reset vector data and then starts the routine after reset.

20.2.4 How to Execute Command

The command execution is performed by writing command sequences to Flash memory with a store instruction. Flash memory executes each automatic operation command according to the combination of input address-es and data. For detail of the command execution, refer to "20.2.5 Command Description".

An execution of store instruction to the Flash memory is called "bus write cycle". Each command consists of some bus write cycles. In Flash memory, when address and data of bus write cycle are performed in the specified order, the automatic command operation is performed. When the cycle is performed in non-specified order, Flash memory stops command execution and returns to the read mode.

If you cancel the command during the command sequence or input a different command sequence, execute the read command or read/reset command. Then Flash memory stops command execution and returns to the read mode. The read command and read/reset command are called "software reset".

When write command sequence ends, the automatic operation starts and FCSR<RDY/BSY> is set to "0". When the automatic operation normally ends, FCSR<RDY/BSY> = "1" is set and Flash memory returns to the read mode.

New command sequences are not accepted during the automatic operation. If you want to stop the command operation, use a hardware reset. In case that the automatic operation abnormally ends (FCSR<RDY/BSY> remains "0"), Flash memory remains locked and will not return to the read mode. To returns to the read mode, use a hardware reset. If the hardware reset stops the command operation, commands are not normally executed.

Notes on the command execution;

1. To recognize command, command sequencer need to be in the read mode before command starting. Confirm FCSR<RDY/BSY> = 1 is set prior to the first bus write cycle of each command. Consecutively, it is recommended that the read command is executed.
2. Execute each command sequence from outside of Flash memory.
3. Execute sequentially each bus write cycle by data transfer instruction in one-word (32-bit).
4. Do not access Flash memory during the each command sequence. Do not generate any interrupt or fault except reset.
5. Upon issuing a command, if any address or data is incorrectly written, make sure to return to the read mode by using software reset.

20.2.5 Command Description

This section explains each command content. For detail of specific command sequences, refer to "20.2.6 Command Sequence".

20.2.5.1 Automatic Page Program

(1) Operation Description

The automatic page program writes data per page. When the program writes data to multiple pages, a page command need to be executed in page by page. Writing across pages is not possible.

Writing to Flash memory means that data cell of "1" becomes data of "0". It is not possible to become data cell of "1" from data of "0". To become data cell of "1" from "0", the erase operation is required.

The automatic page program is allowed only once to each page already erased. Either data cell of "1" or "0" cannot be written data twice or more. If rewriting to a page that has already been written once, the automatic page program is needed to be set again after the automatic block erase or automatic chip erase command is executed.

Note 1: Page program execution to the same page twice or more without erasing operation may damage the device.

Note 2: Writing to the protected block is not possible.

(2) How to Set

The 1st to 3rd bus write cycles indicate the automatic page program command.

In the 4th bus write cycle, the first address and data of the page are written. On and after 5th bus cycle, one page data will be written sequentially. Data is written in one-word unit (32-bit).

If a part of the page is written, set "0xFFFFFFFF" as data, which means not required to write, for entire one page.

No automatic verify operation is performed internally in the device. So, be sure to read the data programmed to confirm that it has been correctly written.

If the automatic page program is abnormally terminated, that page has been failed to write. It is recommended not to use the device or not to use the block including the failed address.

20.2.5.2 Automatic Chip Erase

(1) Operation Description

The automatic chip erase is executed to the memory cell of all addresses. If protected blocks are contained, these blocks will not be erased. If all blocks are protected, the automatic chip erase operation will not performed and will return to the read mode after a command sequence is input.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic chip erase command. After the command sequence is input, the automatic chip erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

20.2.5.3 Automatic Block Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified block. If the specified block is protected, erase operation is not executed.

(2) How to Set

The 1st to 5th bus write cycles indicate the automatic block erase command. In the 6th bus write cycle, the block to be erased is specified. After the command sequence is input, the automatic block erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

20.2.5.4 Automatic Protect Bit Program

(1) Operation Description

The automatic protect bit program writes "1" to a protect bit at a time. To set "0" to a protect bit, use the automatic protect bit erase command.

For detail of the protect function, refer to "20.1.5 Protect/Security Function".

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit program command. In the 7th bus write cycle, the protect bit to be written is specified. After the command sequence is input, the automatic protect bit program starts. Check whether write operation is normally terminated with FCPSRA<BLK1><BLK0>.

20.2.5.5 Auto Protect Bit Erase

(1) Operation Description

The automatic protect bit erase command operation depends on the security status. For detail of security status, refer to "20.1.5 Protect/Security Function".

- Non-security status

  Clear the specified protect bit to "0". Protect bit erase is performed in 4-bit unit.
- Security status

  Erase all protect bits after all addresses of Flash memory are erased.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit erase command. In the 7th bus write cycle, the protect bit to be erased is specified. After the command sequence is input, the automatic protect bit erase operation starts.

In the non-security status, specified protect bit is erased. Check whether erase operation is normally terminated with FCPSRA<BLK1><BLK0>.

In the security status, all addresses and all protect of Flash memory bits are erased. Confirm if data and protect bits are erased normally. If necessary, execute the automatic protect bit erase, automatic chip erase or automatic block erase.

All cases are the same as other commands, FCSR<RDY/BSY> becomes "0" during the automatic protect bit erase command operation. After the operation is complete, FCSR<RDY/BSY> becomes "1" and Flash memory will return to the read mode. To abort the operation, a hardware reset is required.

20.2.5.6 ID-Read

(1) Operation Description

The ID-Read command can read information including Flash memory type and three types of codes such as a maker code, device code and macro code.

(2) How to Set

The 1st to 3rd bus write cycles indicate the ID-Read command. In the 4th bus write cycle, the code to be read is specified. After the 4th bus write cycle, read operation in the arbitrary flash area acquires codes.

The ID-Read can be executed successively. The 4th bus write cycle and reading ID value can be executed repeatedly.

The ID-Read command does not automatically return to the read mode. To return to the read mode, execute the read command, read/reset command or hardware reset.

20.2.5.7 Read Command and Read/reset Command (Software Reset)

(1) Operation Description

A command to return Flash memory to the read mode.

When the ID-Read command is executed, macro stops at the current status without automatically return to the read mode. To return to the read mode from this situation, use the read command or read/reset command. It is also used to cancel the command when commands are input to the middle.

(2) How to Set

The 1st bus cycle indicates the read command. The 1st to 3rd bus write cycles indicate the read/reset command. After either command sequence is executed, Flash memory returns to the read mode.

20.2.6 Command Sequence

20.2.6.1 Command Sequence List

Table 20-5 shows addresses and data of bus write cycle in each command.

All command cycles except the 5th bus cycle of ID-Read command are bus write cycles. A bus write cycle is performed by 32-bit (1-word) data transfer instruction. (Following table shows only lower 8 bits of data.)

For detail of addresses, refer to Table 20-6. Use below values to "command" described in a column of Addr[15:9] in the Table 20-6.

Note 1) Always set to "0" to the address bit [1:0].

Note 2) Set below values to the address bit [19] according to Flash memory size.
Memory size is 1MB or less : Always set to "0"
Memory size is over 1MB : If bus write to 1MB area or less, the bit is set to "0".
If bus write to over 1MB area, the bit is set to "1".

Table 20-5 Command Sequence

Command	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read/reset	0xX55X	0xXAAX	0xX55X	–	–	–	–
	0xAA	0x55	0xF0	–	–	–	–
ID-Read	0xX55X	0xXAAX	0xX55X	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page program	0xX55X	0xXAAX	0xX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Automatic block erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Automatic protect bit program	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protect bit erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- IA: IDAddress
- ID: ID data
- PA: Program page address
- PD: Program data (32-bit data)

  After the 4th bus cycle, input data in the order of the addresses per page
- BA: Block address (see Table 22-7)

- PBA: Protect bit address (see Table 20-8)

20.2.6.2 Address Bit Configuration in the Bus Cycle

Table 20-6 is used in conjunction with "Table 22-5 Command Sequence".

Set the address setting according to the normal bus write cycle address configuration from the first bus cycle.

Table 20-6 Address bit configuration in the bus write cycle

Address	Addr [31:15]	Addr [14]	Addr [13:12]	Addr [11:9]	Addr [8:7]	Addr [6:4]	Addr [3:0]

Normal Command	Normal bus write cycle address configuration				
	Flash area	"0" is recommended.	Command	Addr[1:0] = "0" (fixed) Other bits = "0" (recommended)	
ID-READ	IA: ID address (Setting of the 4th bus write cycle address for ID-READ)				
	Flash area	"0" is recom-mended.	ID Address	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)	
Block erase	BA: Block address(Setting of the 6th bus write cycle address for block erase)				
	Block address (Table 22-7)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)			
Automatic page pro-gram	PA: Program page address (Setting of the 4th bus write cycle address for page program)				
	Page address			Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)	
Protect bit program	PBA: Protect bit address (Setting of the 7th bus write cycle address for protect bit program)				
	Flash area	Fix to "0"		Protect bit selection (Table 22-8)	Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)

20.2.6.3 Block Address(BA)

Table 20-7 shows block addresses. Specify any address included in the block to be erased in the 6th bus write cycle of the automatic block erase command.

Table 20-7 Block address

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
1	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
0	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32

20.2.6.4 How to Specify Protect Bit (PBA)

The protect bit is specified in 1-bit unit in programming and in 4-bit unit in erasing.

Table 20-8 shows a protect bit selection table of the automatic protect bit program. The column of address example indicates an address described in upper side is used in the use boot mode and the lower side is used in the single boot mode.

Four protect bits are erased by the automatic protect bit erase command in all.

Table 20-8 Protect bit program address

Block	Protect bit	Address of 7th bus write cycle			Address example [31:0]
		Address [14:9]	Address [8]	Address [7]	
Block0	<BLK0>	Fix to "0"	0	0	0x0000_0000 0x3F80_0000
Block1	<BLK1>		0	1	0x0000_0080 0x3F80_0080

20.2.6.5 ID-Read Code (IA, ID)

Table 20-9 shows how to specify a code and the content using ID-Read command.

The column of address example indicates an address described in the upper side is used in the use boot mode and the lower side is used in the single boot mode

Table 20-9 ID-Read Command codes and contents

Code	ID[7:0]	IA[13:12]	Address Example [31:0]
Manufacture code	0x98	0b00	0x0000_0000 0x3F80_0000
Device code	0x5A	0b01	0x0000_1000 0x3F80_1000
-	Reserved	0b10	-
Macro code	0x33	0b11	0x0000_3000 0x3F80_3000

20.2.6.6 Example of Command Sequence

(1) use boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x0000_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	–	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_00F0	–	–	–	–
ID-Read	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

(2) Data single boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x3F80_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–	–	–	–
	Data	0x0000_00AA	0x3F80_0055	0x3F80_00F0	–	–	–	–
ID-Read	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

20.2.7 Flowchart

20.2.7.1 Automatic Program

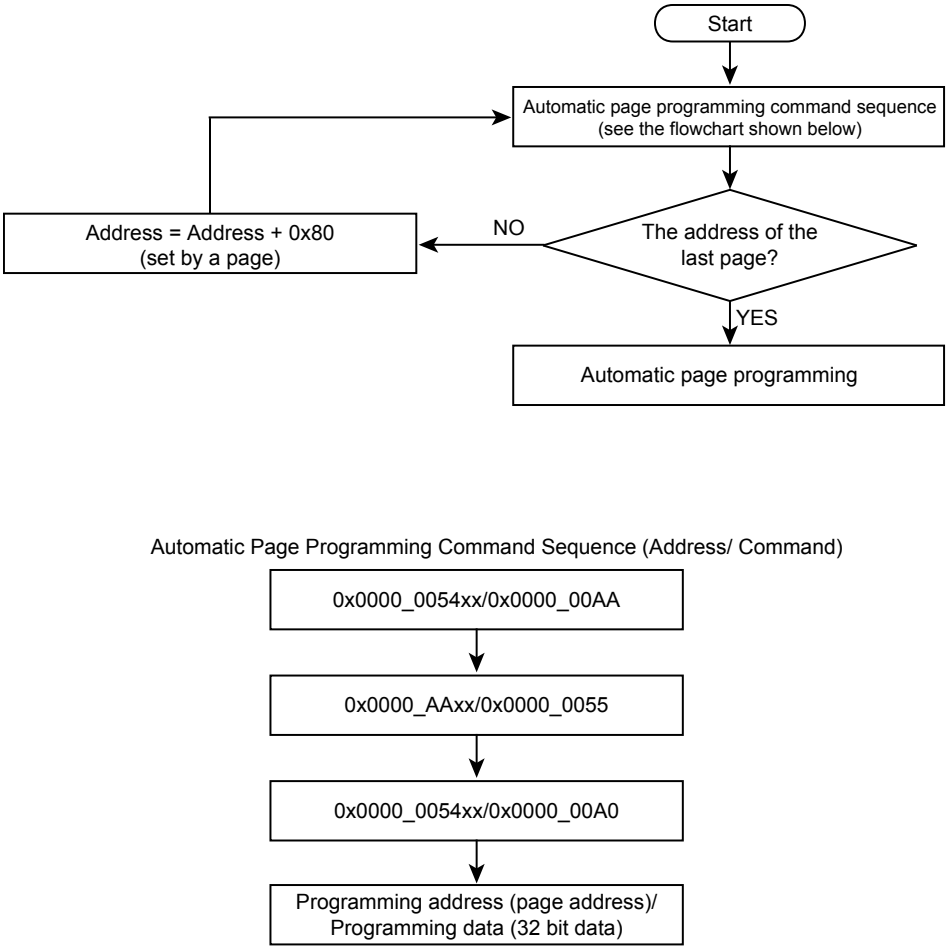


Figure 20-4 Flowchart of automatic program

20.2.7.2 Automatic Erase

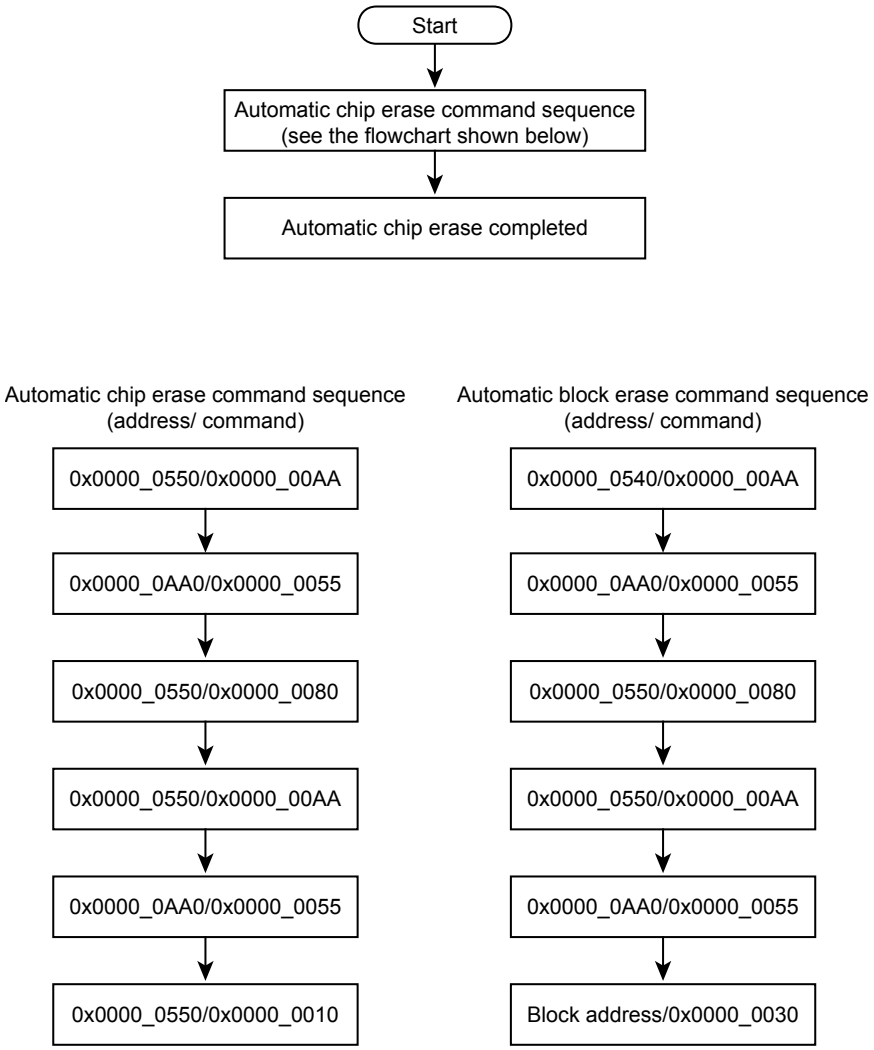


Figure 20-5 Flowchart of automatic erase

20.3 How to Reprogram Flash using Single Boot Mode

The single boot mode utilizes a program contained in built-in BOOT ROM for reprogrammig Flash memory. In this mode, BOOT ROM is mapped to the area containg interrupt vector tables and Flash memory is mapped to another address area other than BOOT ROM area.

In the boot mode, Flash memory is reprogrammed using serial command/data transfer. With connecting serial channel (UART) of this device to the external host, a reprogramming program is copied from the external host to the built-in RAM. A reprogramming routine in the RAM is executed to reprogram Flash memory. For details of communication with host, follow the protocol described later.

Even in the single boot mode, do not generate interrupt/fault except reset to avoid abnormal program termination.

To secure the contents of Flash memory in the single chip mode (normal operation mode), once re-programming is complete, it is recommended to protect relevant flash blocks against accidental erasure during subsequent single chip operations.

20.3.1 Mode Setting

In order to execute the on-board programming, this device is booted-up in the single boot mode. Below setting is for the single boot mode setting.

$\overline{BOOT}$ = 0
$\overline{RESET}$ = 0 → 1

While $\overline{BOOT}$ pin is set to the above in advance, set $\overline{RESET}$ pin to "0". Then release $\overline{RESET}$ pin, the device will boot-up in the single boot mode.

Note: This device may start up in single-boot mode, when the $\overline{BOOT}$ pin is "Low" level at power-on. Therefore when MCU starting in single mode, The $\overline{BOOT}$ pin must be at "High" level at power-on until reset release operation is completed.

20.3.2 Interface Specification

This section describes UART communication format in the single boot mode. The serial operation supports UART (asynchronous communication) modes. In order to execute the on-board programming, set the communication format of the programming controller as well.

- UART communication

  Communication channel: channel 0

  Serial transfer mode: UART (asynchronous), half-duplex, LSB first

  Data length: 8-bit

  Parity bit: None

  STOP bit: 1-bit

  Baud rate: Arbitrary baud rate

The boot program operates the clock/mode control block setting as an initial condition. For detail of the initial setting of the clock, refer to "Clock/Mode control".

As explained in the "20.3.5.1 Serial Operation Mode Determination", a baud rate is determined by the 16-bit timer (TMRB). When determining the baud rate, communication is executed by 1/16 of a desired baud rate. Therefore, the communication baud rate must be within the measurable range. The timer count clock op-erates at ΦT1 (fc/2).

A handshaking pin of I/O interface mode outputs "Low" waiting in receive state and outputs "High" in transmission state. Check the handshaking pin before communications and must follow the communication protocol.

Table 20-10 shows the pins used in the boot program. Other than these pins are not used by the boot program.

Table 20-10 Pin connection

Pin		Interface
		UART
Mode setting pin	$\overline{\text{BOOT}}$	o
Reset pin	$\overline{\text{RESET}}$	o
Communication pin	TXD0 (PE0)	o
	RXD0 (PE1)	o

o:used ×:unused

20.3.3 Restrictions on Internal Memories

Note that the single boot mode places restrictions on the built-in RAM and built-in flash memory as shown in Table 20-11.

Table 20-11 Restrictions on the memories in the single boot mode

Memory	Restrictions
Internal RAM	Boot program uses the memory as a work area through 0x2000_0000 to 0x2000_03FF. Store the program 0x2000_0400 through the end address of RAM. The start address of the program must be even address.
Internal flash memory	The following addresses are assigned for storing software ID information and passowerds. Storing program in below addresses is not recommendable. 0x3F80_FFF0 to 0x3F80_FFFF

Note: If a password is erased data (0xFF), it is difficult to protect data secure due to an easy-to-guess password. Even if the single boot mode is not used, it is recommended to set a unique value as a password.

20.3.4 Operation Command

The boot program provides the following operation commands.

Table 20-12 Operation command data

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory chip erase and protect bit erase

20.3.4.1 RAM Transfer

The RAM transfer is to store data from the controller to the built-in RAM . When the transfer is complete normally, a user program starts. User program can use the memory address of 0x2000_0400 or later except 0x2000_0000 to 0x2000_03FF for the boot program. CPU will start execution from RAM store start address. The start address must be even address.

This RAM transfer function enables user-specific on-board programming control. In order to execute the on-board programming by a user program, use Flash memory command sequence explained in 20.2.6.

20.3.4.2 Flash Memory Chip Erase and Protect Bit Erase

Flash memory chip erase and protect bit erase commands erase the entire blocks of Flash memory and write/erase protects of all blocks regardless of write/erase protect or security status.

20.3.5 Common Operation regardless of Command

This section describes common operation under the boot program execution.

20.3.5.1 Serial Operation Mode Determination

When the controller communicates via UART, set the 1st byte to 0x86 at the desired baud rate. Figure 20-6 shows waveforms in each case.

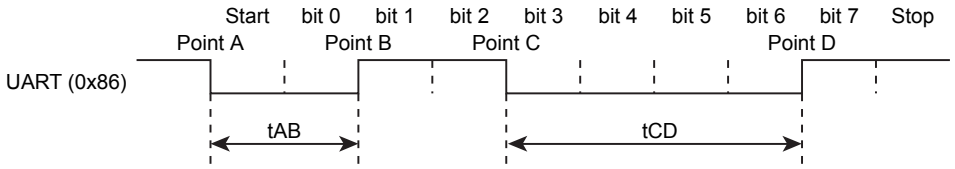


Figure 20-6 Serial operation mode determination data

Figure 20-7 shows a flowchart of boot program. Using 16-bit timer (TMRB) with the time of tAB, tAC and tAD, the 1st byte of serial operation mode determination data (0x86) after reset is provided. In Figure 20-7, the CPU monitors level of the receive pin, and obtains a timer value at the moment when the receive pin's level is changed. Consequently, the timer values of tAB, tAC and tAD have a margin of error. In addition, note that if the transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin.

The flowchart in Figure 20-8 shows the serial operation mode is determined that the time length of the receive pin is long or short. If the length is tAB ≤ tCD, the serial operation mode is determined as UART mode. The time of tAD is used whether the automatic baud rate setting is enable or not. Note that timer values of tAB, tAC and tAD have a margin of error. If the baud rate is high and operation frequency is low, each timer value becomes small. This may generates unexpected determination occurs. (To prevent this problem, re-set UART within the programming routine.)

For example, When UART mode is utilized, the controller should allow for a time-out period where the time is expected to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time.

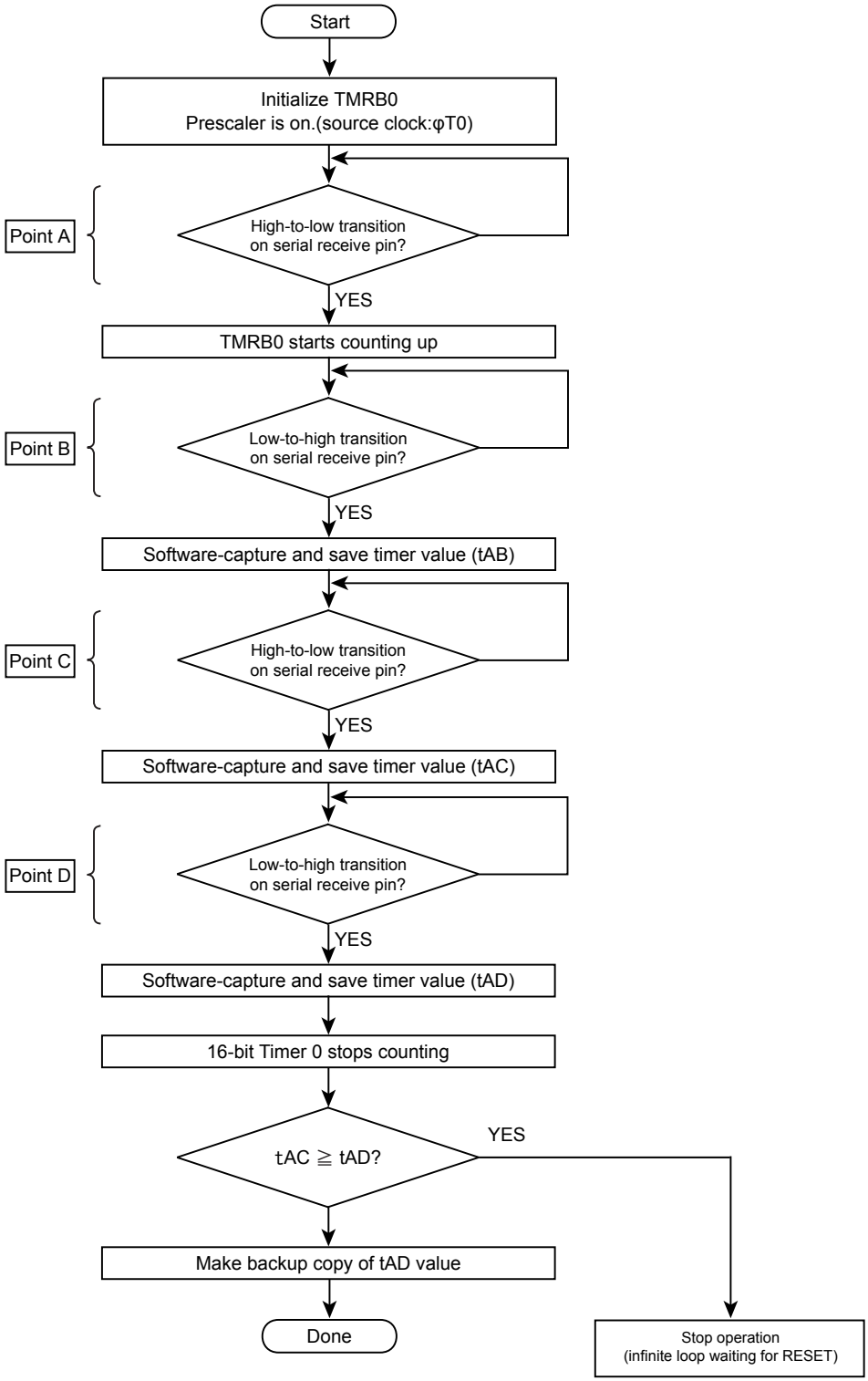


Figure 20-7 Serial operation mode receive flowchart

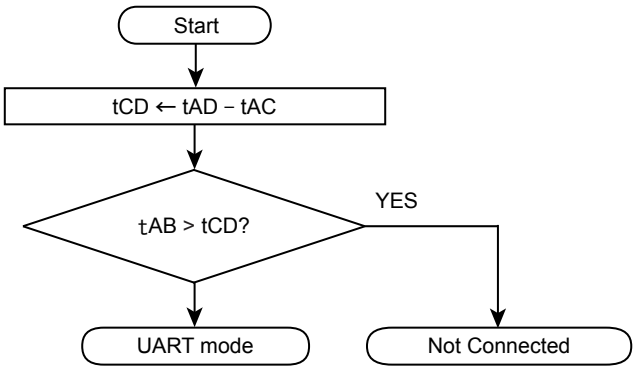


Figure 20-8 Serial operation mode determination flowchart

20.3.5.2 Acknowledge Response Data

The boot program represents processing states in specific codes and sends them to the controller. Table 20-13 to Table 20-16 show the values of acknowledge responses to each receive data.

In Table 20-14 to Table 20-16, the upper four bits of the acknowledge response are equal to those of the operation command data. The 3rd bit indicates a receive error. The 0th bit indicates an invalid operation command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not performed in I/O Interface mode.

Table 20-13 ACK response to the serial operation determination data

Transmit data	Description
0x86	Determined that UART communication is possible. (Note)

Note: When the serial operation is determined as UART, if the baud rate setting is determined as unacceptable, the boot program aborts without sending back any response.

Table 20-14 ACK response to the operation command data

Transmit data	Description
0x?8 (Note)	A receive error occurrs in the operation command data
0x?1 (Note)	An undefined operation command data is received normally.
0x10	Determined as a RAM transfer command
0x40	Determined as a flash memory chip erase command

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 20-15 ACK response to the CHECK SUM data

Transmit data	Description
0xN8 (Note)	A receive error occurs.
0xN1 (Note)	A CHECK SUM or a password error occurs.
0xN0 (Note)	The CHECK SUM value is correct.

Note: The upper 4 bits of the ACK response data are the same as those of the operation command data.

Table 20-16 ACK response to Flash memory chip erase and protect bit erase operation

Transmit data	Description
0x54	Determined as a erase enable command
0x4F	Erase command is complete.
0x4C	Erase command is abnormally terminated.

Note: Even when an erasecommand is performed normally, a Negatice acknowledge may be returned by ACK response. Check the FCSR<RDY/BSY> to make sure the command sequence end , and then hold for 200 μ s or more, after that reconfirm the erase status.

20.3.5.3 Password Determination

The boot program use the below area to determine whether a password is required or use as a password.

Area	Address
Password requirement determination	0x3F80_FFF0 (1byte)
Password area	0x3F80_FFF4 to 0x3F80_FFFF (12byte)

The RAM Transfer command performs a password verification regardless of necessity judging data. Flash memory chip erase or protect bit erase command performs a password verification only when necessity judging is determined as "required".

Password requirement setting	Data
Need password	Other than 0xFF
No password	0xFF

If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

(1) Password verification using RAM transfer command

If all these address locations contain the same bytes of data other than 0xFF, this condition is determined as a password area error as shown in Figure 22-9. In this case, the boot program returns an error acknowledge (0x11) in response to the 17th byte of checksum value regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

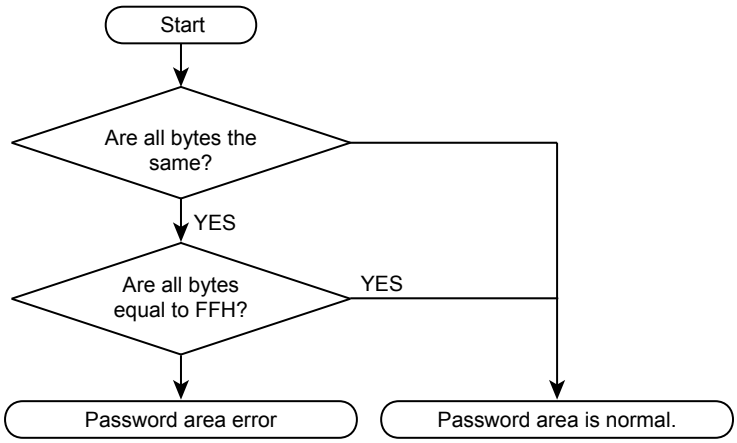


Figure 20-9 Password area check flowchart

(2) Password verification to Flash memory chip erase and protect bit erase command

When a password is enable in the erase password necessity determination area as shown in Figure 20-10 and the passwords are identical data, a password area error occurs. If a password area error is determined, an ACK response to the 17th byte of CHECK SUM sends 0x41 regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

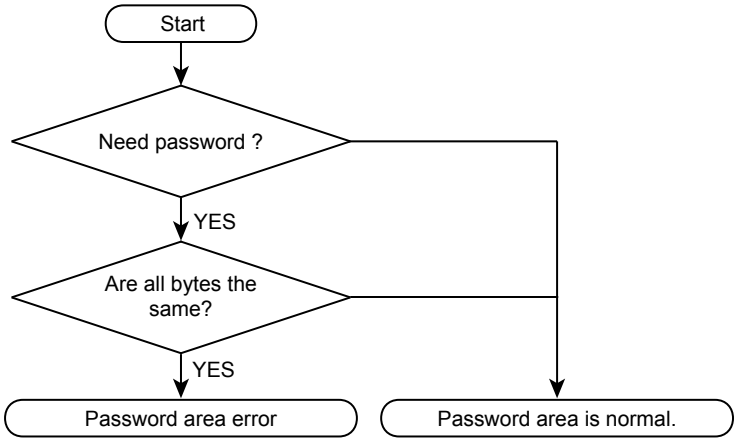


Figure 20-10 Password area check flowchart

20.3.5.4 CHECK SUM Calculation

The checksum is calculated by 8-bit addition to transmit data, dropping the carries, and taking the two's complement of the total sum. The controller must perform the same checksum operation in transmitting checksum bytes.

Example of CHECK SUM

To calculate the checksum for a series of 0xE5 and 0xF6, perform 8-bit addition.

0xE5 + 0xF6 = 0x1DB

Take the two’s complement of the sum to the lower 8-bit, and that is a checksum value. So the boot program sends 0x25 to the controller.

0 − 0xDB = 0x25

20.3.6 Transfer Format at RAM Transfer

This section shows a RAM transfer command format. Transfer directions in the table are indicated as follows: Transfer direction (C→T): Controller to TMPM3U0FSDMG

Transfer direction (C←T): TMPM3U0FSDMG to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determina-tion, refer to "20.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, the program determines whether a baud setting is possible. If not, the program stops and communication is shutdown.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
3	C→T	Operation command data (0x10)	Sends RAM transfer command data (0x10).
4	C←T	ACK response to operation command Normal state: 0x10 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 20-12, receive data is echoed back. In the case of RAM transfer, 0x10 is echoed back and the transfer data branches to the RAM transfer service routine. If the data does not correspond to the command in Table 20-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.)
5 to 16	C→T	Password data (12-byte) 0x3F80_FF04 to 0x3F80_FF0F	Checks data in the password area. For detail of password area checking, refer to "20.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F80_FFF0 to 0x3F80_FFFF of data of Flash memory. If the data does not match the address, a password error flag is set.
17	C→T	5th to 16th byte of CHECK SUM values	Send 5th to 16th byte of CHECK SUM values. For detail of CHECK SUM calculation, refer to 20.3.5.4.

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 5th to 17th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 17th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x11 that means a password error and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
19	C→T	RAM store start Address 31 to 24	Sends a start address of block transfer for RAM store. The 19th byte corresponds to 31st to 24th bit of address.The 22nd byte corresponds to 7th to 0th bit of address. Specify the address to the address 0x2000_0400 through the last address of RAM. The address must be even address.
20	C→T	RAM store start Address 23 to 16	
21	C→T	RAM store start Address 15 to 8	
22	C→T	RAM store start Address 7 to 0	
23	C→T	Number of RAM store bytes 15 to 8	Set the number of bytes to perform block transfer. The 23rd byte corresponds to the15th bit to 8th bit of transfer bytes. The 24th byte corresponds to 7th bit to 0th bit of transfer bytes. Specify the data to be stored in the address from 0x2000_0400 through the last address of RAM.
24	C→T	Number of RAM store bytes 7 to 0	
25	C→T	19th to 24th byte of CHECK SUM value	Send 19th byte to 24th byte of CHECK SUM values
26	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 19th byte to 25th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 25th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
27 to m	C→T	RAM stored data	Sends same bytes of data specified in 23th bytes to 24 byte for RAM stored data.
m+1	C→T	27 to m byte of CHECK SUM value	Sends 27th byte to m byte of CHECK SUM value
m+2	C←T	ACK response to CHECK SUM value Normal state:0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 27th byte to m+1 byte of receive data have errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Then checks m+1 byte of CHECK SUM data, if errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
–	–	–	If m + 2nd byte of ACK response data is normal ACK response data, the transfer data branches to the address specified in 19th byte to 22 byte.

20.3.7 Transfer Format of Flash memory Chip Erase and Protect Bit Erase

This section shows a transfer format of Flash memory chip erase and protect bit erase commands. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM3U0FSDMG

Transfer direction (C←T): TMPM3U0FSDMG to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determina-tion, refer to"20.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, checks if baud rate setting can be done. If not, operation stops communications.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
3	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
4	C←T	ACK response to the operation command Normal state: 0x40 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 20-12, receive data is echoed back. If the data does not correspond to the command in Table 20-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command. (3rd byte) Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
5 to 16	C→T	Password data (12-byte) 0x3F80_FF04 to 0x3F80_FF0F	If password necessity is set to "none", this data is dummy data. If password necessity is set to "necessary", checks data in the password area. For a method of password area data checking, refer to "20.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F80_FFF0 to 0x3F80_FFFF of data of Flash memory in order. If the data does not match, a password error flag is set.
17	C→T	5 to 16 th byte CHECK SUMvalue	Sends 5th byte to 16 byte of CHECK SUM value. For a method of CHECK SUM calculation, refer to"20.3.5.4 CHECK SUM Calculation".

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to the CHECK SUM value Normal state: 0x40 Abnormal state: 0x41 Communication error: 0x48	If password necessity is set to "none", sends a normal ACK response data 0x40. If password necessity is set to "necessary", first checks if receive errors exist in the 5th byte to 17th byte receive data. (UART mode only) If a receive error exists, sends a ACK response data 0x48 that means abnormal communications and waits for next operation command. (3rd byte) Then checks 17th byte of CHECK SUM data. If error occurs, sends 0x41 and waits for a next operation command (3rd byte) Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x41that means a password error and waits for a next operation command (3rd byte) If all procedure normally ends, sends a normal ACK response data 0x40.
19	C→T	Erase enable command data (0x54)	Sends an enable command data (0x54).
20	C←T	ACK response to the erase enable command Normal state: 0x54 Abnormal state: 0xX1 Communication error: 0x58	First, checks if 19th byte of receive data has errors. If receive errors exist, sends a ACK response data (bit 3) 0x58 that means abnormal communication and waits for next operation command (3rd byte). Then, if 19th byte of receive data corresponds to the erase enable command, receive data is echoed back (normal ACK response data). In this case, 0x54 is echoed back and the transfer data branches into Flash memory chip erase process routine. If the data does not correspond to the erase enable command, sends a ACK response data (bit 0) 0xX1 and waits for next operation command. Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
21	C→T	ACK response to the erase command (note1) Normal state: 0x4F Abnormal state: 0x4C	If the operation is normally complete, the end code (0x4F) is returned. If erase error occurs, an error code (0x4C) is returned.
–	–	–	Waits for a next operation command.

Note 1: Even when an erasecommand is performed normally, a Negatice acknowledge may be returned by ACK response. Check the FCSR<RDY/BSY> to make sure the command sequence end , and then hold for 200 μ s or more, after that reconfirm the erase status.

20.3.8 Boot Program Whole Flowchart

This section shows a boot program whole flowchart.

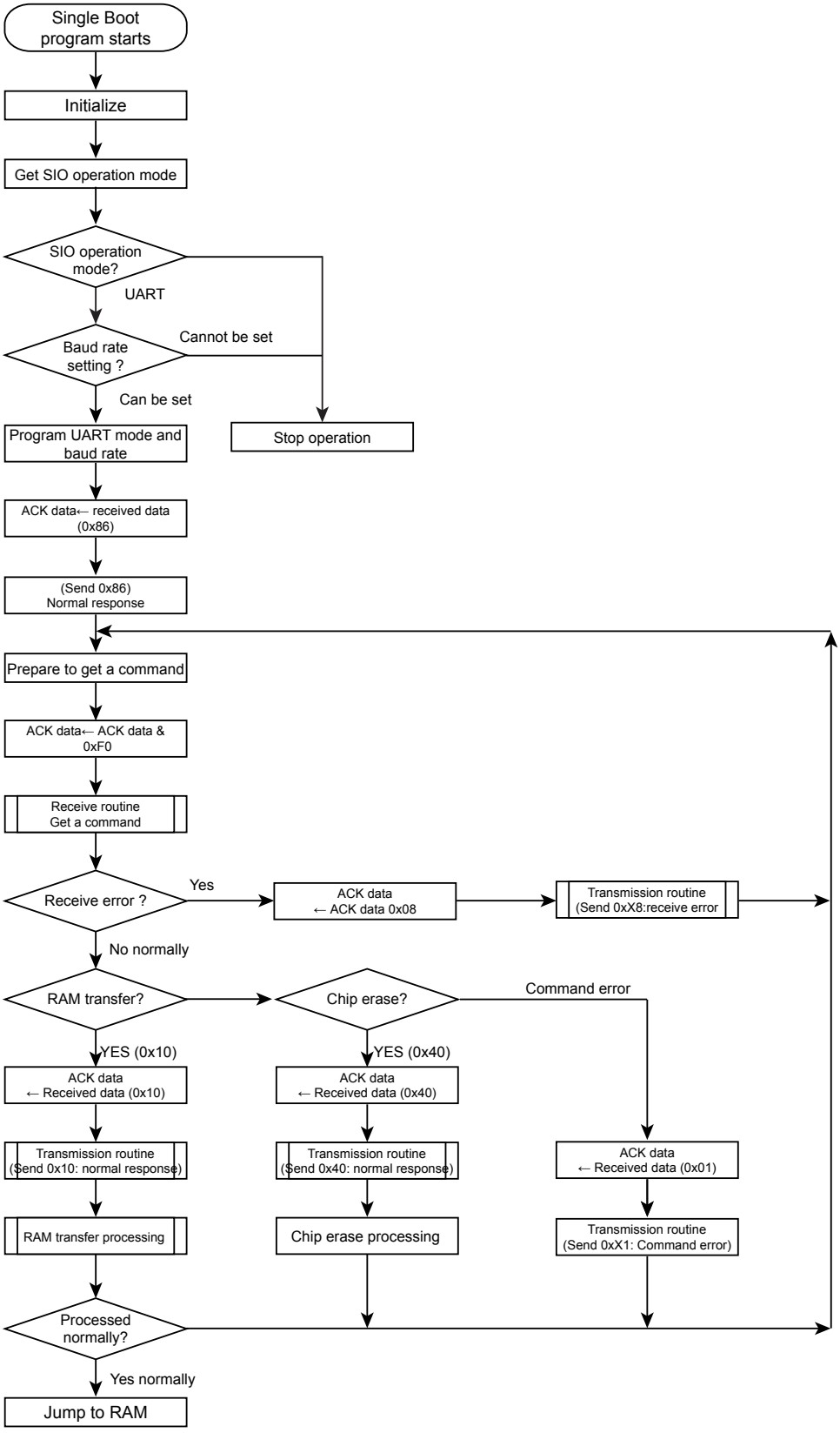


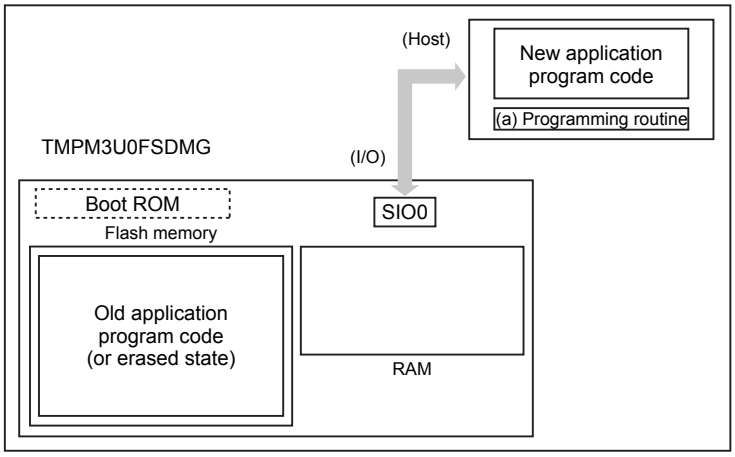
Figure 20-11 Boot program whole flowchart

20.3.9 Reprogramming Procedure of Flash using reprogramming algorithm in the on-chip BOOT ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip boot ROM.

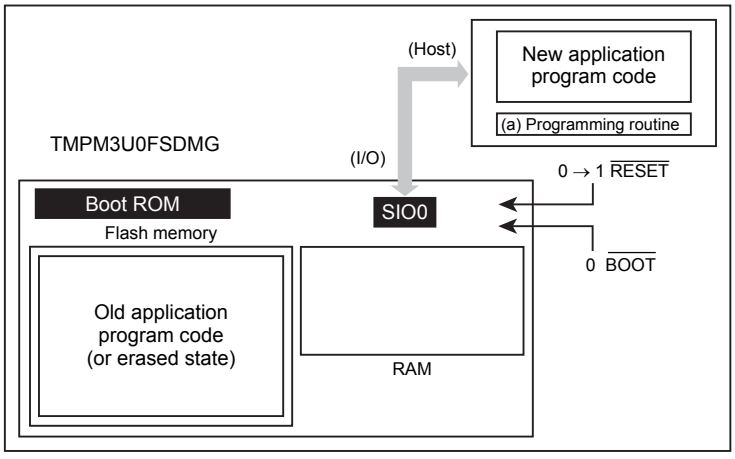
20.3.9.1 Step-1

The condition of Flash memory does not care whether a user program made of former versions has been written or erased. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to an external host. A programming routine (a) is prepared on the host.



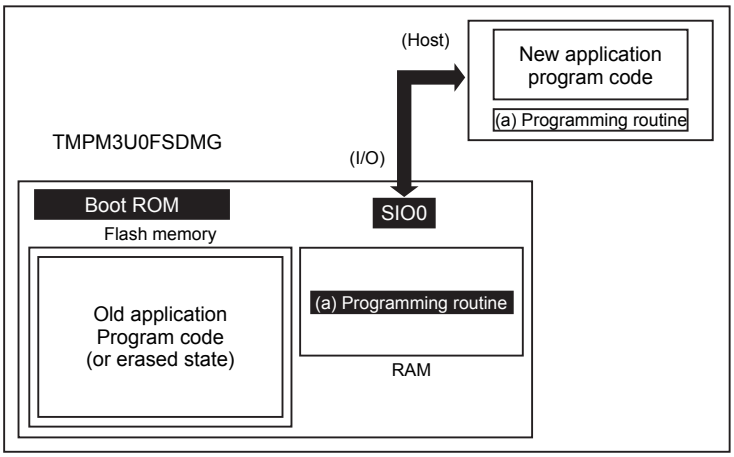
20.3.9.2 Step-2

Release the reset by pin condition setting in the boot mode and boot-up the BOOT ROM. According to the procedure of boot mode, transfer the programming routine (a) via SIO0 from the source (host). A password verification with the the password in the user application program is perfomed. (If Flash memory is erased, an erase data (0xFF) is dealt with a password.)



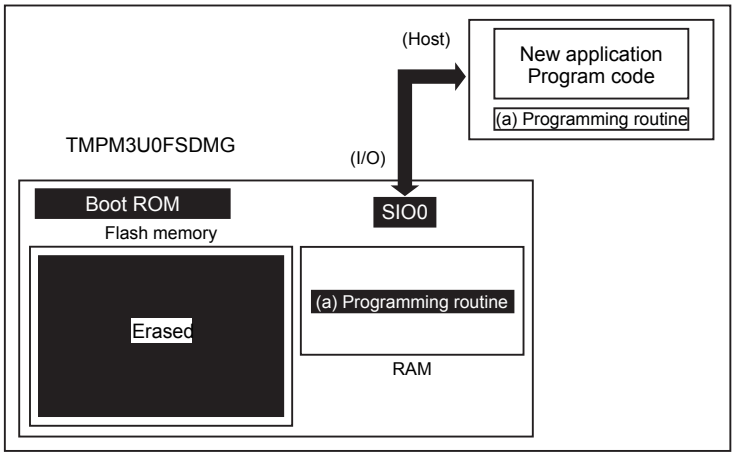
20.3.9.3 Step-3

If the password verification is complete, the boot program transfer a programming routine (a) from the host into the on-chip RAM. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



20.3.9.4 Step-4

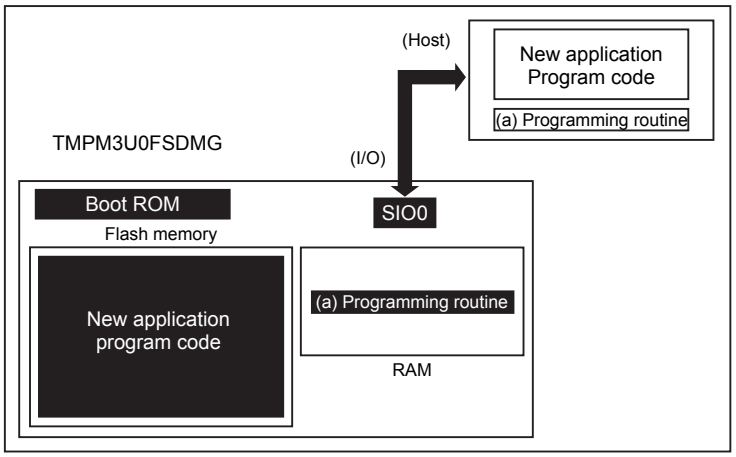
The boot program jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing old application program codes. The Block Erase or Chip Erase command is be used.



20.3.9.5 Step-5

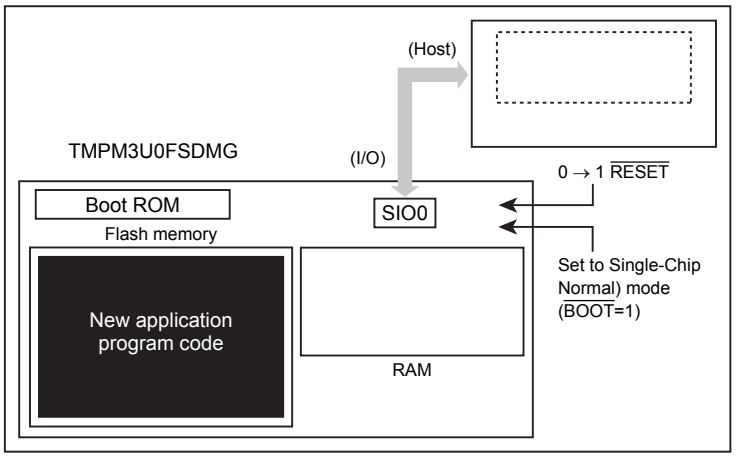
The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is complete, the writing or erase protection of that flash area in the user's program must be set.

In the example below, new program code comes from the same host via the same 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 a hardware board and programming routine to suit your particular needs.



20.3.9.6 Step-6

When programming of Flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the device re-boots in the single-chip (Normal) mode to execute the new program.



20.4 Programming in the User Boot Mode

A user Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the user application is different from the serial I/O. It operates in the single chip mode; therefore, a switch from normal mode in which user application is activated in the use boot mode to the user boot mode for programming flash is required. Specifically, add a mode judgment routine to the reset service routine in the user application program.

The condition to switch the modes needs to be set according to the user's system setup condition. Also, a flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to the user boot mode. The data in built-in Flash memory cannot be read out during erase/reprogramming mode. Thus, reprogramming routine must be take place while it is stored in the area outside of Flash memory area. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental reprogramming. Be sure not to generate interrupt/fault except reset to avoid abnormal termination during the user boot mode.

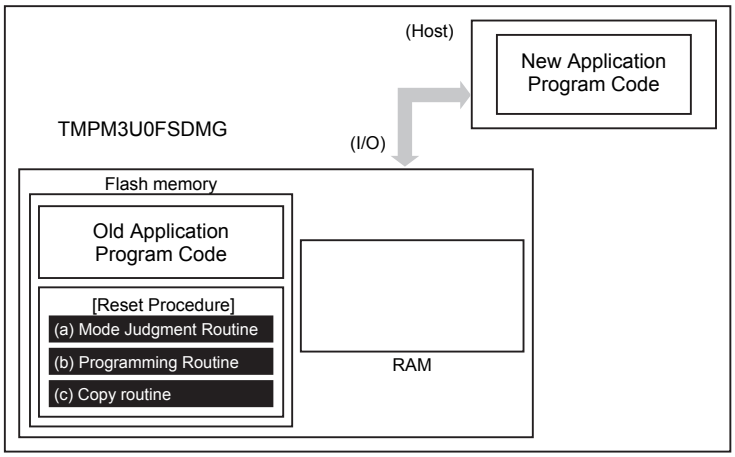
Taking examples from two cases such as the method that reprogramming routine stored in Flash memory (1-A) and transferred from the external device (1-B), the following section explains the procedure. For a detail of the program/erase to Flash memory, refer to "20.2 Detail of Flash Memory".

20.4.1 (1-A) Procedure that a Programming Routine Stored in Flash memory

20.4.1.1 Step-1

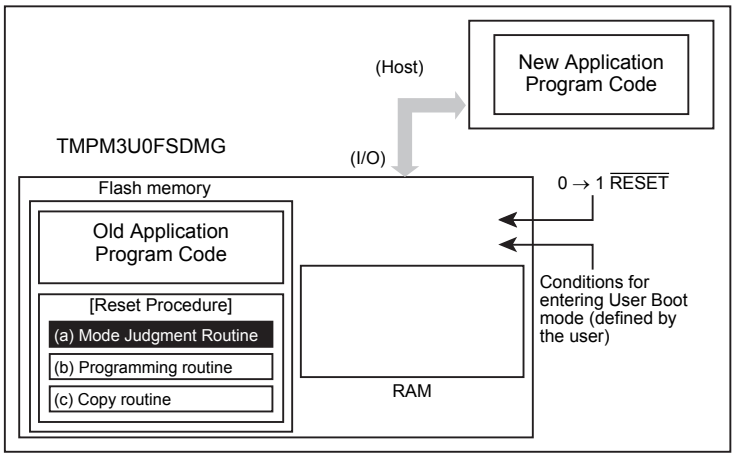
A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following three program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to user boot mode or not
(b) Flash programming routine:	A program to download new program from the host controller and re-program Flash memory
(c) Copy routine:	A program to copy the data described in (a) to the built-in RAM or external memory device



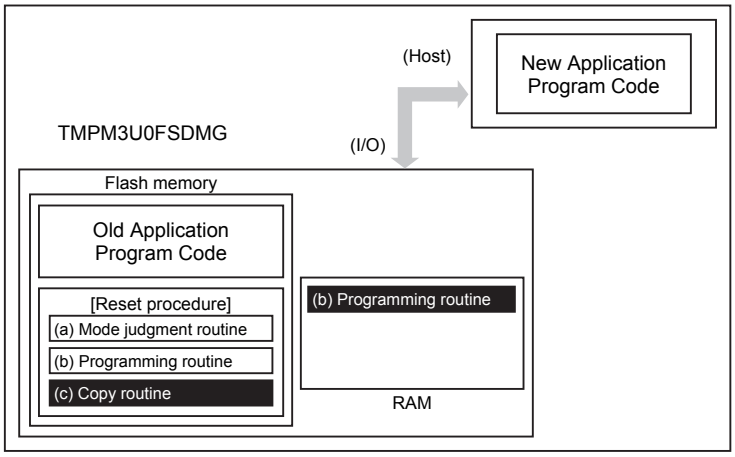
20.4.1.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.



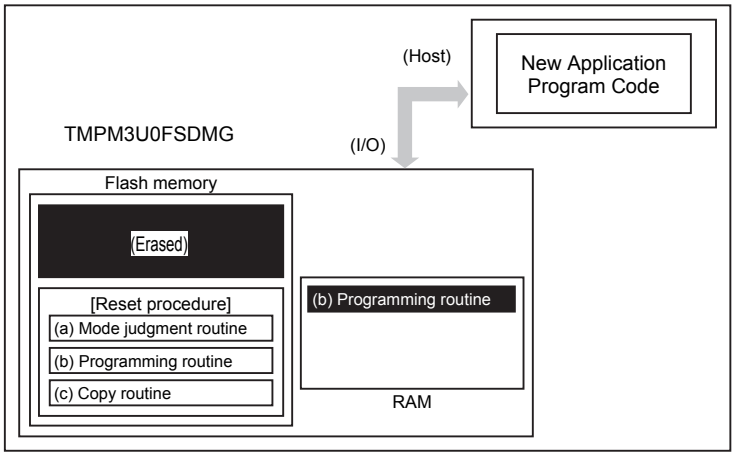
20.4.1.3 Step-3

Once the device enters the user boot mode, execute the copy routine (C) to download the flash programming routine (b) from the host controller to the built-in RAM .



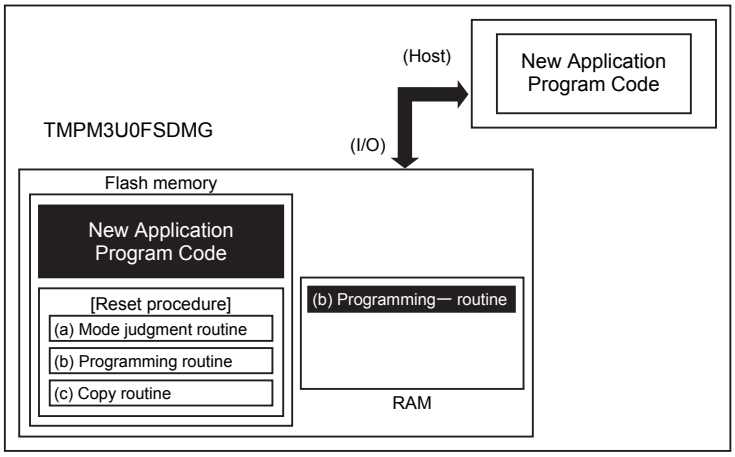
20.4.1.4 Step-4

Jump to the the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



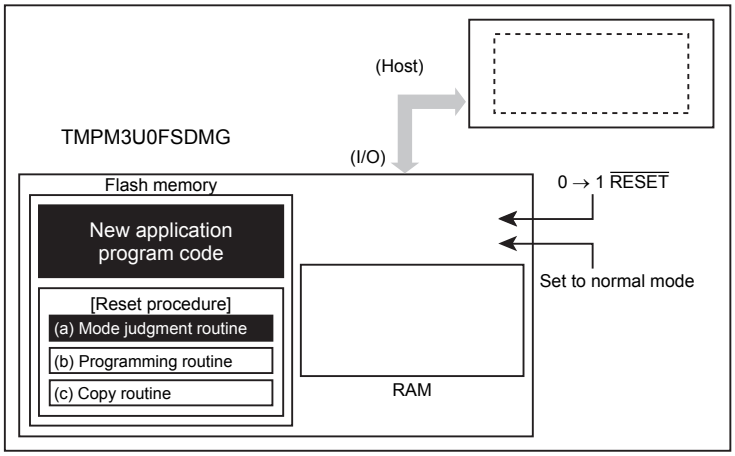
20.4.1.5 Step-5

Continue to execute the flash programming routine to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



20.4.1.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



20.4.2 (1-B) Procedure that a Programming Routine is transferred from External Host

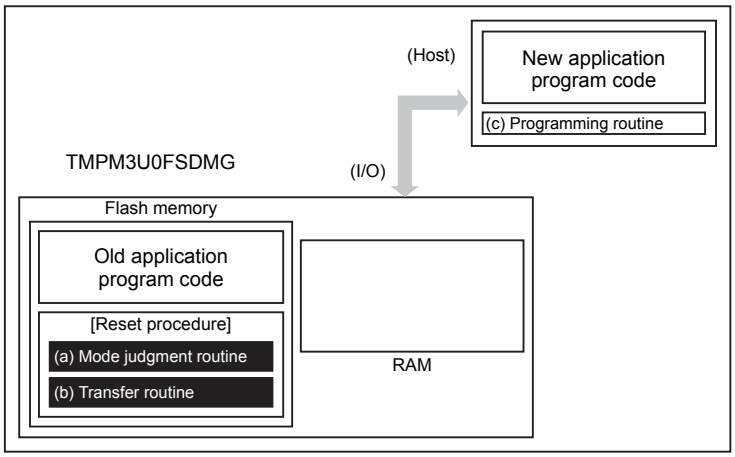
20.4.2.1 Step-1

A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following two program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to reprogramming operation
(b) Transfer routine:	A program to obtain a reprogramming program from the external device.

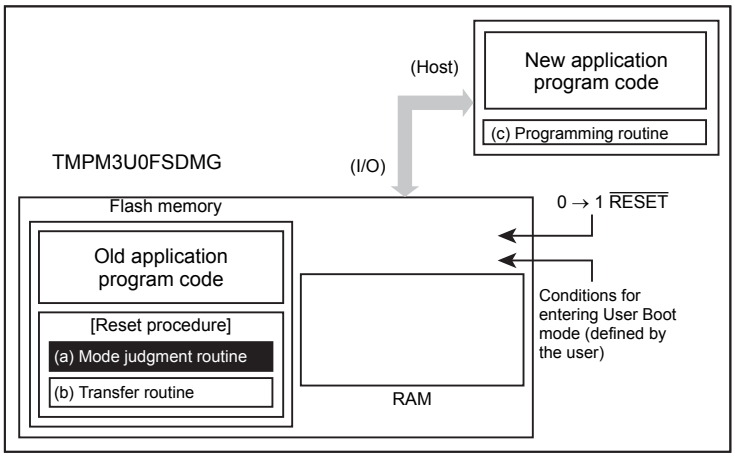
In addition, prepare a reprogramming routine shown below must be stored on the host controller.

(c) Reprogramming routine:	A program to reprogram data



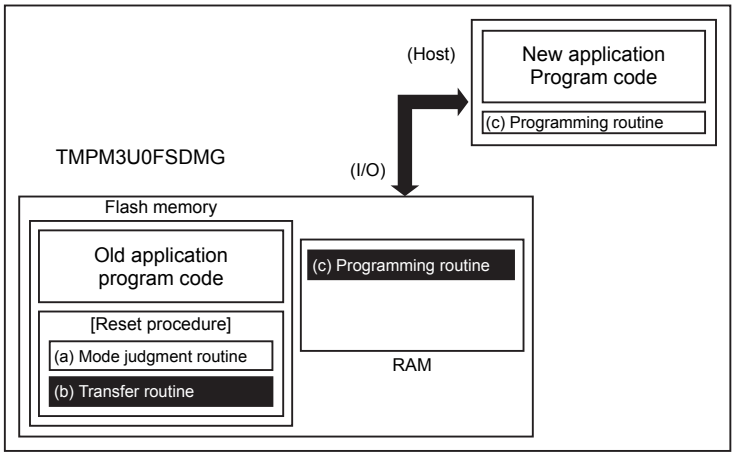
20.4.2.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.



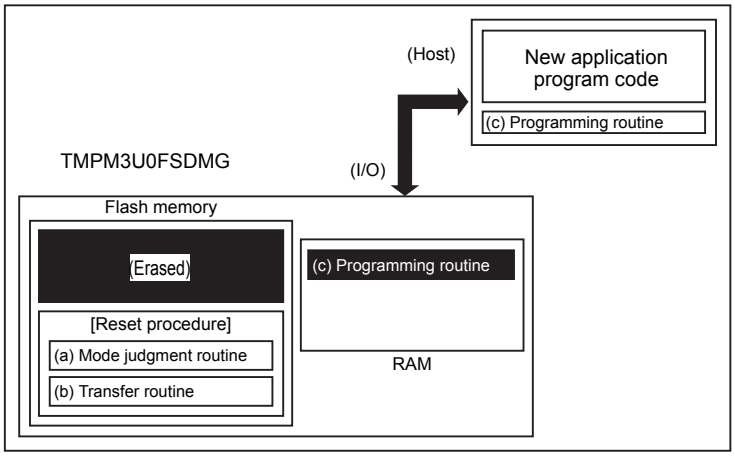
20.4.2.3 Step-3

Once the device enters the user boot mode, execute the transfer routine (b) to download the programming routine (c) from the host controller to the built-in RAM .



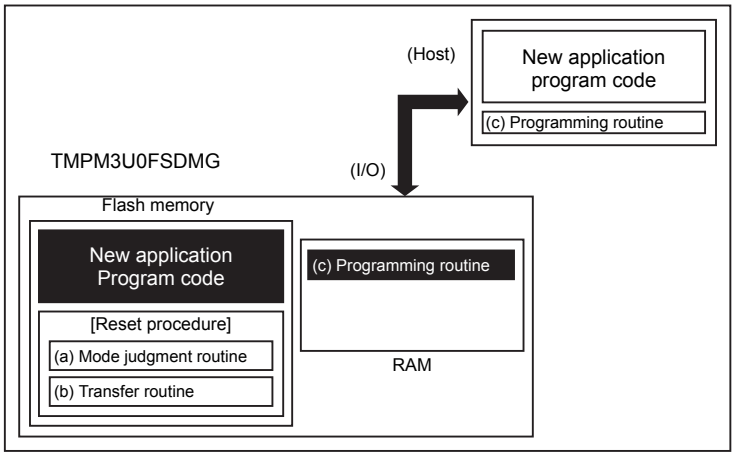
20.4.2.4 Step-4

Jump to the the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



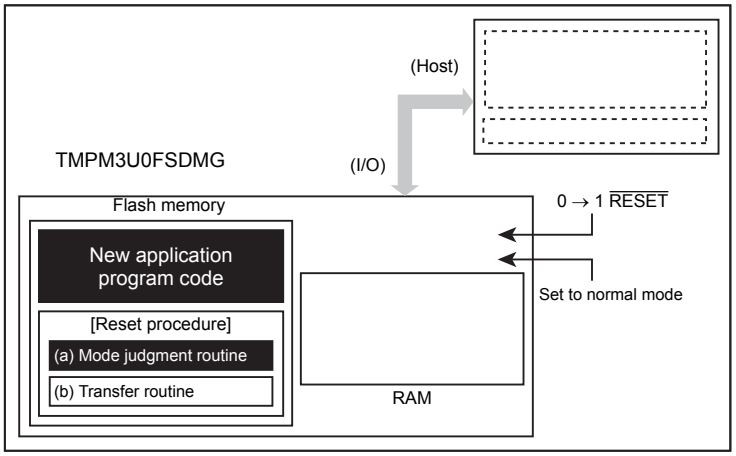
20.4.2.5 Step-5

Continue to execute the flash programming routine (c) to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



20.4.2.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



21. Debug Interface

21.1 Specification Overview

The TMPM3U0FSDMG contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the Debug interface and the Embedded Trace Macrocell™ (ETM) unit for trace output. Trace data is output to the dedicated pins (SWV) via the on-chip Trace Port Interface Unit (TPIU).For more informarion of SWJ-DP,ETM and TPIY , please refer to the Arm manual "Cortex-M3 Technical Reference Manual" issued by Arm Limited.

21.2 Features of SWJ-DP

SWJ-DP supports the two-pin Serial Wire Debug Port (SWDCK, SWDIO) and the JTAG Debug Port (TDI, TDO, TMS, TCK).

21.3 Features of ETM

ETM supports trace output from SWV.

21.4 Pin Functions

The debug interface pins can also be used as general-purpose ports. The PB3 and PB4 are shared between the JTAG debug port function and the serial wire debug port function. The PB5 is shared between the JTAG debug port function and the SWV trace output function.

Table 21-1 SWJ-DP, ETM function

SWJ-DP Pin name	Name of port	JTAG debug function		SW debug	
		I/O	Description	I/O	Description
TMS/SWDIO	PB3	Input	JTAG Test Mode Selection	I/O	Serial Wire Data Input/Output
TCK/SWCLK	PB4	Input	JTAG Test Check	Input	Serial Wire Clock
TDO/SWV	PB5	Output	JTAG Test Data Output	(Input) (Note1)	(Serial Wire Viewer Output)
TDI	PB6	Input	JTAG Test Data Input	-	-

Note: In case of enabling SWV function

After reset, the PB3, PB4, PB5 and PB6 are configured as debug port function pins. The functions of other debug interface pins need to be programmed as required. Debug interface pins can use general purpose port that is not use debug interface.

Table 21-2 below summarizes the debug interface pin functions and related port settings after reset.

Table 21-2 Debug interface pins and port setting after reset

Initial Setting	Port (Bit name)	Debug Function	Port Setting After Reset (-;No register)					
			Function (PBFR)	Input (PBIE)	Output (PBCR)	Open Drain (PBOD)	Pull-up (PBPUP)	Pull-down (PBPDN)
DEBUG	PB3	TMS/SWDIO	1	1	1	0	1	0
DEBUG	PB4	TCK/SWCLK	1	1	0	0	0	1
DEBUG	PB5	TDO/SWV	1	0	1	0	0	0
DEBUG	PB6	TDI	1	1	0	0	1	0

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

Note 1: If PB3 and PB5 are configured as debug function pins, output continues to be enabled even in STOP mode regardless of the setting of the CGSTBYCR<DRVE>.

Note 2: If PB4 is configured as a debug function pin, it prevents a low power consumption mode from being fully effective. Configure PB4 to function as a general-purpose port if the debug function is not used.

21.5 Connection with a Debug Tool

21.5.1 How to connect

For how to connect a debug tool, refer to the method recommended by each manufacture.Debug interface pins have pull-up or pull-down register. When connect with pull-up or pull-down riggers, be sure their settings.

21.5.2 When use general purpose port

When debugging, do not change setting debug interface to general purpose port by program. Then, MCU will be unable to control signals received from the debugging tools and can not continue debugging. According to the usage of the debug interface pins, be sure their setting.

Table 21-3 Debug Interface

Usage	Using Debug Interface (O:Enable, -:Disable)			
	TDI	TDO/SWV	TCK/ SWCLK	TMS/ SWDIO
JTAG+SW (After RESET)	O	O	O	O
JTAG+SW (non $\overline{\text{TRST}}$)	O	O	O	O
JTAG+TRACE	O	O	O	O
SW	-	-	O	O
SW+SWV	-	O	O	O
Disable Debug function	-	-	-	-

21.6 Peripherals operation during HALT mode

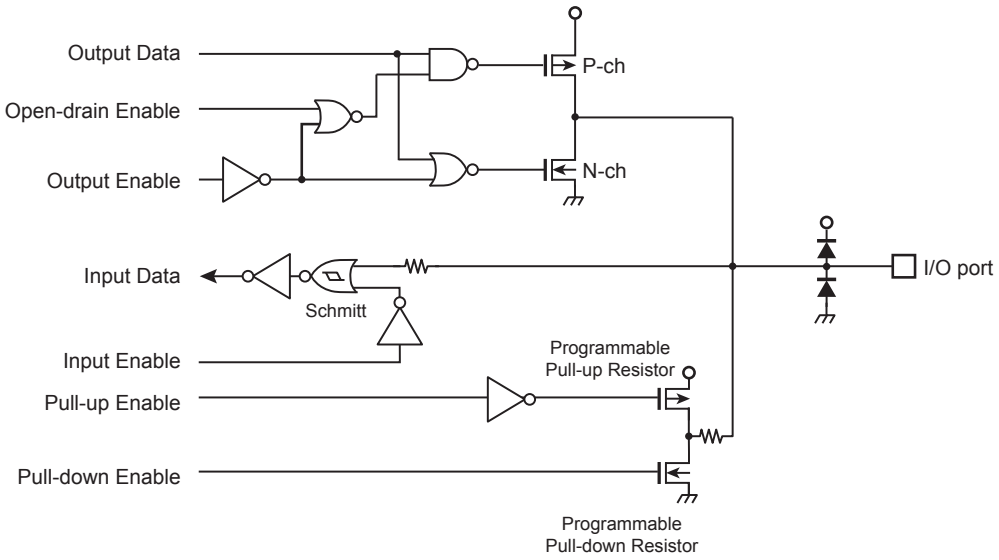
When Break during debugging, Cortex-M3 CPU core going into HALT mode. Watch dog timer (WDT) is stopped counting automatically. And 16bit timer/counter can specify the status (continue operating or stop) in HALT mode. Other peripherals are continue operating.

22. Port Section Equivalent Circuit Schematic

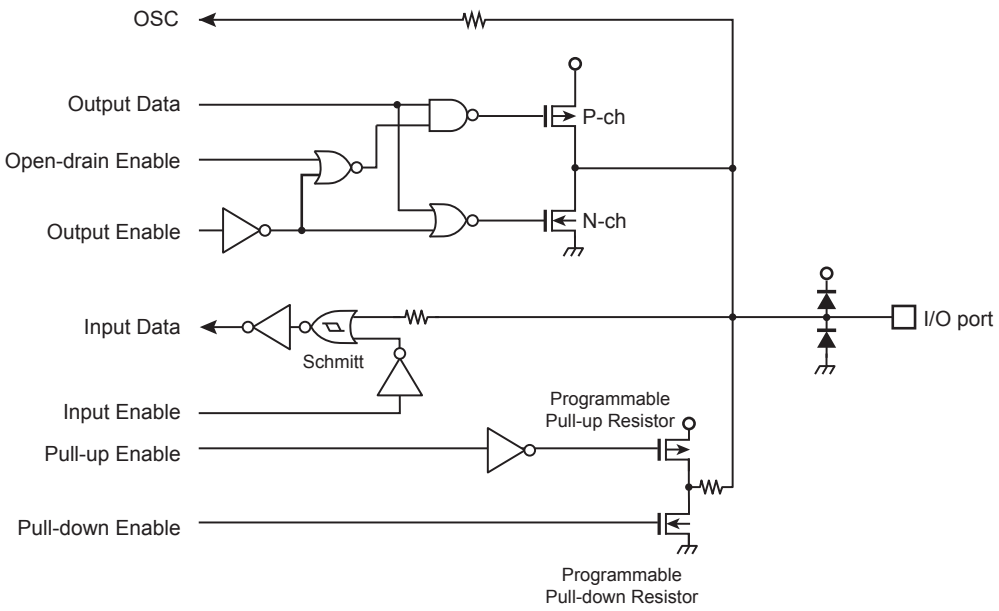
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of Ω to several hundreds of Ω. Feedback resistor and Damping resistorare shown with a typical value.

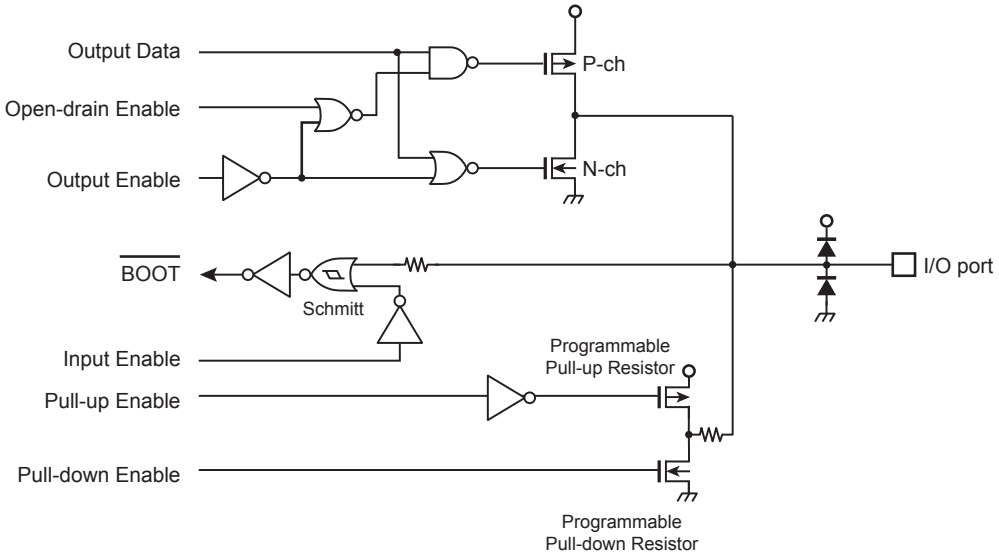
22.1 PB3 to 6, PE0 to 2, PG0 to 6



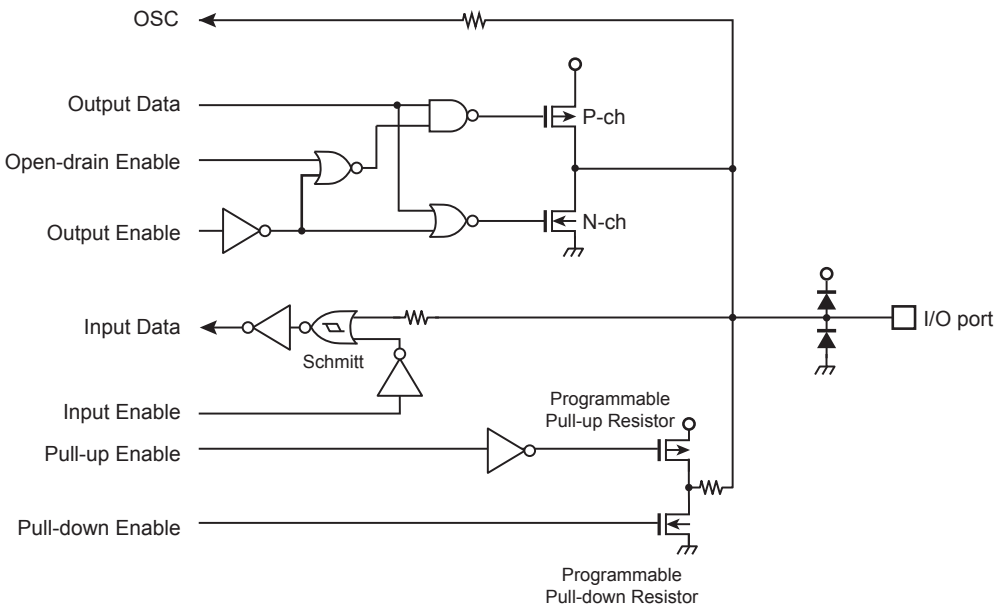
22.2 PJ6 to 7, PK0 to 1



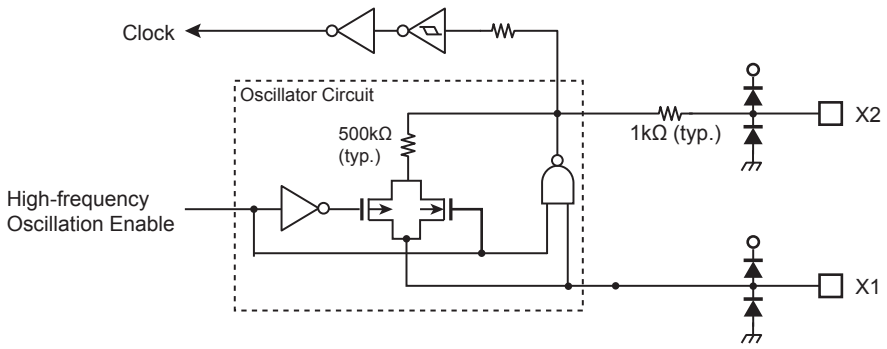
22.3 PF0



22.4 PM0 to 1

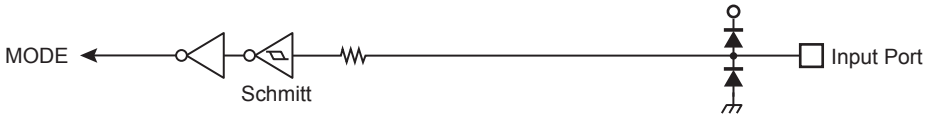


22.5 X1, X2



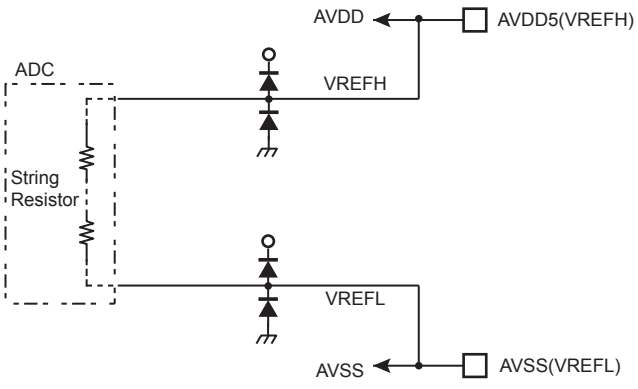
22.6 $\overline{\text{RESET}}$

22.7 MODE

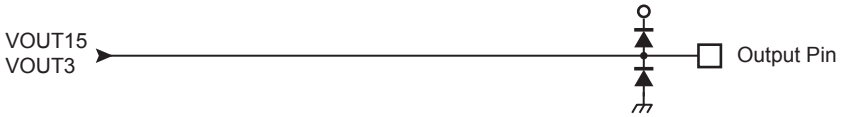


(Note)MODE pin is fixed to GND.

22.8 VREFHB, VREFLB



22.9 VOUT15, VOUT3



23. Electrical Characteristics

23.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		VDD (Note2)	−0.3 to 6	V
Capacitor voltage		VOUT15	−0.3 to 3	V
		VOUT3	−0.3 to 3.9	
Input voltage		V_{IN}	−0.3 to VDD + 0.3 (Note2)	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	50	
Power consumption		PD	350	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−55 to 125	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	−40 to 105	°C
	During Flash W/E		0 to 70	

Note1: **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.**

Note2: **VDD = DVDD5B / AVDD5B**

23.2 DC Electrical Characteristics (1/2)

DVSS = DVSSB = AVSSB = 0V, Ta = −40 to 105 °C

Parameter		Symbol	Rating		Min.	Typ. (Note 1)	Max.	Unit
Supply voltage (Note 2)	DVDD5B AVDD5B	VDD	f_{OSC} = 8 or 10 MHz fsys = 1 to 40 MHz		4.5	−	5.5	V
Supply voltage (during Flash W/E) (Note 2)	DVDD5B AVDD5B	VDD	f_{OSC} = 8 or 10 MHz fsys = 1 to 40 MHz (Ta (°C) = 0 to 70)		4.5	−	5.5	V
Supply voltage (Power-on or Power-off) (Note 5)	DVDD5B AVDD5B	VDD	f_{OSC} = 8 or 10 MHz fsys = 1 to 40 MHz		3.9	−	5.5	V
Low-level input voltage	Schmitt-Input	V_{IL1}	VDD = 4.5V to 5.5V (Note 4)		−0.3	−	0.25 VDD	V
High-level input voltage	Schmitt-Input	V_{IH1}	VDD = 4.5V to 5.5V (Note 4)		0.75VDD		VDD+0.3	V
Capacitance for VOUT15 and VOUT3 (Note 3)		C_{out}	VOUT15, VOUT3		3.3	−	4.7	μF
Low-level output voltage		V_{OL}	I_{OL} = 1.6 mA	VDD ≥ 4.5V (Note 4)	−	−	0.4	V
High-level output voltage		V_{OH}	I_{OH} = −1.6 mA	VDD ≥ 4.5V (Note 4)	4.1	−	−	V
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ VDD (Note 4)		−	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ VDD -0.2 (Note 4)		−	0.05	±10	
Pull-up resister at Reset		R_{RST}	4.5 ≤ VDD ≤ 5.5 (Note 4)		−	50	150	kΩ
Programmable pull-up/pull-down resistor		P_{KH}	4.5 ≤ VDD ≤ 5.5 (Note 4)		−	50	150	kΩ
Schmitt-Triggered port		V_{TH}	4.5 ≤ VDD ≤ 5.5 (Note 4)		0.3	0.6	−	V
Pin capacitance (Except power supply pins)	Digital pins	C_{IO1}	fc = 1 MHz		−	−	10	pF
	Analog pins PJ6, PJ7, PK0 and PK1	C_{IO2}			−	−	30	

Note 1: Ta = 25 °C, DVDD5B = AVDD5B = 5V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD5B and AVDD5B.

Note 3: VOUT15 and VOUT3 pin should be connected to GND via same value of capacitance. The IC outside can not have the power supply from VOUT15 and VOUT3.

Note 4: VDD = DVDD5B = AVDD5B

Note 5: It is a voltage range in the case of Power-on or Power-off (when VLTD disabled). In the range whose Power-line is 3.9V ≤ VDD < 4.5V, does not guarantee a 12-bit A/D converter and AC electrical Characteristics.

23.3 DC Electrical Characteristics (2/2)

DVDD5B = AVDD5B = 4.5 V to 5.5 V, Ta = −40 to 105 °C

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2) Gear 1/1	I_{DD}	fsys = 40 MHz	−	15	25	mA
IDLE (Note 3) Gear 1/1			−	5	15	
STOP		−	−	0.1	3	

Note 1: Ta = 25 °C, DVDD5B = AVDD5B = 5V, unless otherwise noted.

Note 2: I_{DD} NORMAL: All functions operates excluding ADC.

Note 3: I_{DD} IDLE : All peripheral functions stopped.

23.4 12-bit ADC Electrical Characteristics

DVDD5B = AVDD5B / VREFHB = 4.5 V to 5.5 V

DVSSB = AVSSB / VREFLB = 0V, Ta = −40 to 105 °C

Parameter		Symbol	Rating	Min.	Typ.	Max	Unit
Analog reference voltage (+)		VREFHB	−	−	AVDD	−	V
Analog input voltage		VAIN	−	AVSS	−	AVDD	V
Supply current	A/D conversion	−	Include IREF	−	6.0	10.0	mA
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 μF Conversion time ≥ 2 μs	−	−	± 8	LSB
DNL error				−	−	± 5	
Offset error				−	−	± 5	
Full-scale error				−	−	± 6	
Total error				−	−	−7 to +11	

Note1: 1LSB = (AVDD − AVSS)/4096 [V]

Note2: AVDD = AVDD5B, AVSS = AVSSB

Note3: The characteristic is measured under the condition in which the only ADC is operating.

23.5 AC Electrical Characteristics

23.5.1 AC measurement condition

AC measurement condition

- Output levels: High = 0.8 × VDD / Low = 0.2 × VDD
- Input levels: Refer to low-level input voltage and high-level input voltage in DC Electrical Characteristics.
- Load capacity:CL=30pF

Note: VDD = DVDD5B = AVDD5B

23.5.2 Serial Channel Timing (SIO)

23.5.2.1 I/O Interface mode

In the table below, the letter x represents the period of the system clock (fsys). It varies depending on the programming of the clock gear function.

(1) SCLK input mode

[Data Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	100	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	100	−	
SCLK cycle	t_{SCY}	8x	−	200	−	
Valid Data Input ← SCLK rise or fall (Note1)	t_{SRD}	30	−	30	−	
SCLK rise or fall → Input Data hold (Note 1)	t_{HSR}	x + 30	−	55	−	

[Data Output]

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	120 (Note 3)	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	120 (Note 3)	−	
SCLK cycle	t_{SCY}	8x	−	240	−	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	0 (Note 2)	−	
SCLK rise or fall → Output Data hold (Note 1)	t_{OHS}	$t_{SCY}/2$	−	120	−	

Note 1: SCLK rise or fall: Measured relative to the programmed active edge of SCLK.
Note 2: A calculated value should use it the SCLK cycle of the range which is not subtracted.
Note 3: t_{OSS} shows the minimum which is not subtracted.

(2) SCLK Output mode

[Data Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
SCLK cycle (programmable)	t_{SCY}	4x	−	100	−	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 30$	−	20	−	
SCLK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$	−	20	−	
Valid Data Input ← SCLK rise	t_{SRD}	45	−	45	−	
SCLK rise → Input Data hold	t_{HSR}	0	−	0	−	

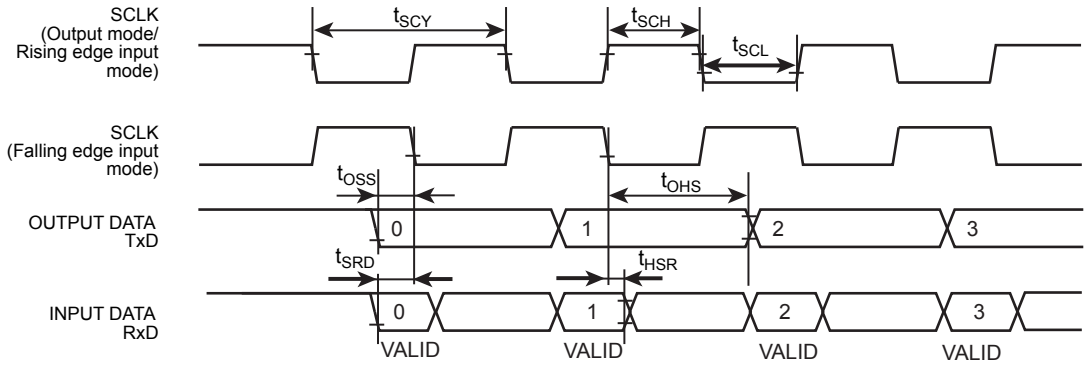


Figure 23-1 Serial channel timing(SIO)

23.5.3 Serial Bus Interface (I2C/SIO)

23.5.3.1 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	$t_{SU;\ STA}$	(Note 5)	−	4.7	−	0.6	−	μs
Data hold time (Input) (Note 3) (Note 4)	$t_{HD;\ DAT}$	−	−	0.0	−	0.0	−	μs
Data setup time	$t_{SU;\ DAT}$	−	−	250	−	100	−	ns
Setup time for a STOP condition	$t_{SU;\ STO}$	−	−	4.0	−	0.6	−	μs
Bus free time between stop condition and start condition	t_{BUF}	(Note 5)	−	4.7	−	1.3	−	μs

Note 1: SCL clock Low width (output) = $(2^{n-1} + 58)/x$

Note 2: SCL clock High width (output) = $(2^{n-1} + 14)/x$

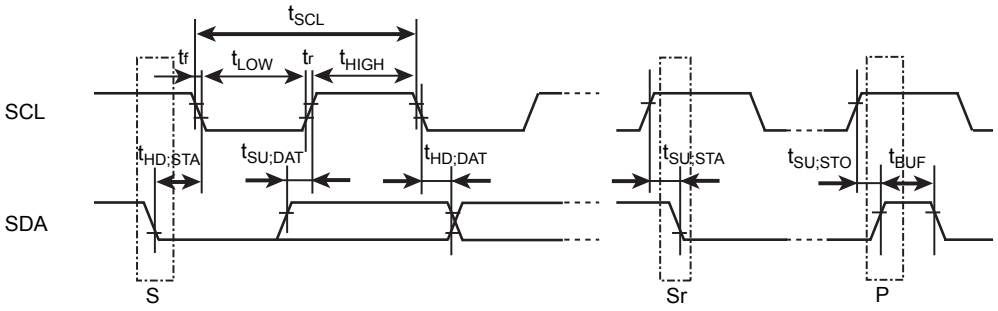
On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400kHz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

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: Re-start condition
P: Stop condition

Figure 23-2 Serial Bus timing (I2C)

23.5.3.2 Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x represents the I2C/SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCK Input Mode (SCK signal with a 50% duty cycle)

Parameter		Symbol	Equation		40 MHz		Unit
			Min.	Max	Min.	Max	
SCK Clock High width (input)		t_{SCH}	4x	−	100	−	ns
SCK Clock Low width (input)		t_{SCL}	4x	−	100	−	
SCK cycle		t_{SCY}	$t_{SCH} + t_{SCL}$	−	200	−	
Output Data	←SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$	−	−20 (Note)	−	
SCK rise	→Output Data hold	t_{OHS}	$t_{SCY}/2 + x$		125	−	
Valid Data input	←SCK rise	t_{SRD}	30 − x		5	−	
SCK rise	→Input Data hold	t_{HSR}	30	−	30	−	

Note: Keep this value positive by adjusting SCK cycle.

(2) SCK Output Mode (SCK signal with a 50% duty cycle)

Parameter		Symbol	Equation		40 MHz		Unit
			Min.	Max	Min.	Max	
SCK cycle (programmable)		t_{SCY}	16x	−	400	−	ns
Output Data	←SCK rise	t_{OSS}	$t_{SCY}/2 - 30$	−	170	−	
SCK rise	→Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$	−	170	−	
Valid Data input	←SCK rise	t_{SRD}	45	−	45	−	
SCK rise	→Input Data hold	t_{OHS}	0	−	0	−	

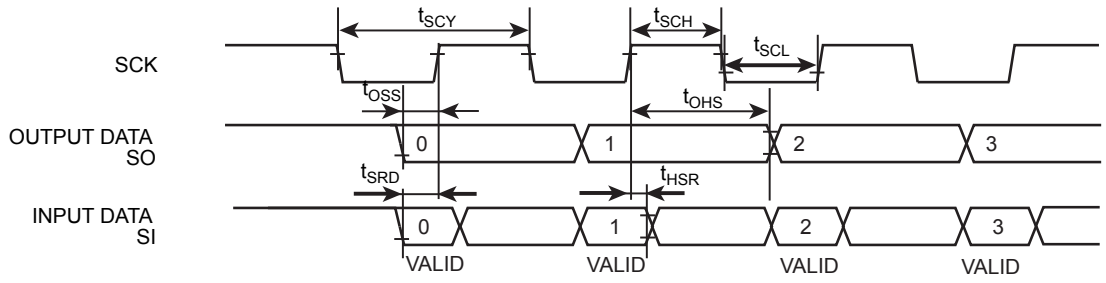


Figure 23-3 Serial Bus timing (SIO)

23.5.4 Event Counter

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
Clock low pulse width	t_{VCKL}	2x + 100	−	150	−	ns
Clock high pulse width	t_{VCKH}	2x + 100	−	150	−	ns

23.5.5 Capture

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
Low pulse width	t_{CPL}	2x + 100	−	150	−	ns
High pulse width	t_{CPH}	2x + 100	−	150	−	ns

23.5.6 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP release interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
Low pulse width for INT6,7 and C	t_{INTAL}	x + 100	−	125	−	ns
High pulse width for INT6,7, and C	t_{INTAH}	x + 100	−	125	−	ns

2. STOP Release Interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min.	Max	Min.	Max	
Low pulse width for INT6,7 and C	t_{INTBL}	100	−	100	−	ns
High pulse width for INT6,7 and C	t_{INTBH}	100	−	100	−	ns

23.5.7 Debug Communication

23.5.7.1 AC measurement condition

- Output levels : High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Load capacitance : CL(TRACECLK) = 25pF,CL(TRACEDATA) = 20pF

23.5.7.2 SWD Interface

Parameter	Symbol	Min.	Max	Unit
CLK cycle	T_{dck}	100	−	ns
DATA hold after CLK rising	T_{d1}	4	−	
DATA valid after CLK rising	T_{d2}	−	45	
DATA valid to CLK rising	T_{ds}	20	−	
DATA hold after CLK falling	T_{dh}	15	−	

23.5.7.3 JTAG Interface

Parameter	Symbol	Min.	Max	Unit
CLK cycle	T_{dck}	100	−	ns
DATA hold after CLK falling	T_{d3}	4	−	
DATA valid after CLK falling	T_{d4}	−	45	
DATA valid to CLK rising	T_{ds}	20	−	
DATA hold after CLK rising	T_{dh}	15	−	

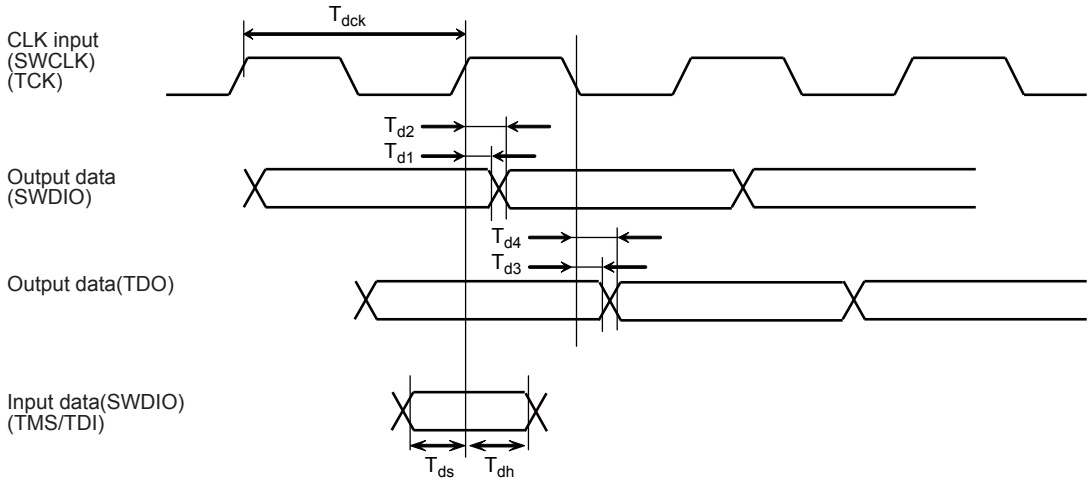


Figure 23-4 JTAG and SWD communication timing

23.5.8 Flash Characteristics

Parameter	Rating	Min.	Typ.	Max	Unit
Guarantee on Flash-memory rewriting	Ta = 0 to 70°C VDD5B = AVDD5B =4.5 to 5.0	-	-	100	times

23.5.9 On chip Oscillator

Parameter	Symbol	Rating	Min.	Typ.	Max	Unit
Oscillation frequency	fosc2	Ta = -40 to 105°C	9.4	9.7	10	MHz

Note: Factory default value

23.5.10 External Oscillator

Parameter	Symbol	Rating	PLL	Min.	Typ.	Max	Unit
High frequency Oscillation	fosc1	Ta = -40 to 105°C	5	7.92	8	8.08	MHz
			4	9.9	10	10.1	

23.6 Oscillation Circuit

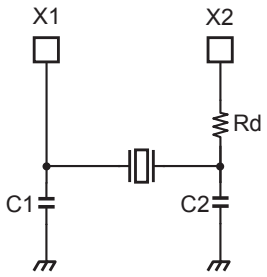


Figure 23-5 High-frequency oscillation connection

Note 1: The load value of the oscillator is the sum of loads (C1 and C2) and the floating load of the actual assembled board. There is a possibility of operating error when using C1 and C2 values in the table below. When designing the board, design the minimum length pattern around the oscillator. We also recommend that oscillator evaluation be carried out using the actual board.

Note 2: Do not be driven X1/X2 by external driver.

The TX03 has been evaluated by the oscillator vender below. Use this information when selecting external parts.

23.6.1 Recommended ceramic oscillator

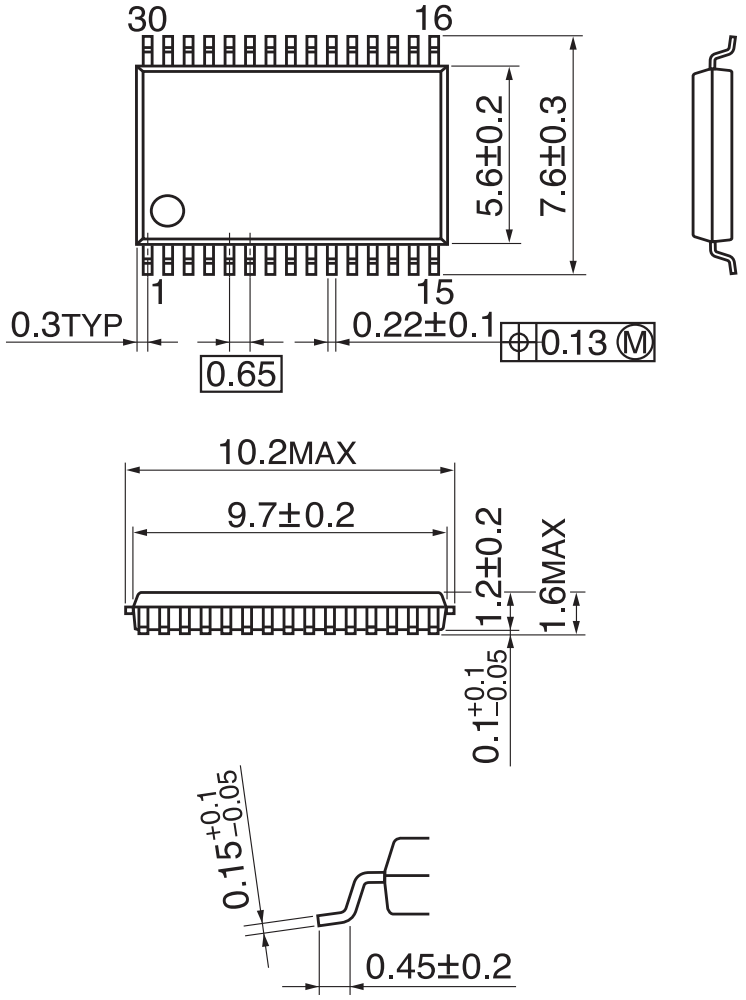
The TX03 recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.

Please refer to the campany's Website for details.

24. Package Dimensions

Type : SSOP30-P-300-0.65

Unit : mm



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.**