TOSHIBA

32 Bit RISC Microcontroller

TX03 Series

TMPM3U6FY/FWFG
TMPM3U6FY/FWDFG

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

General precautions on the use of Toshiba MCUs

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

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

1. The MCUs' operation at power-on

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

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

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

2. Unused pins

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

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

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

3. Clock oscillation stability

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

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

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

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

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

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

Base Address = 0x0000_0000

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

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

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

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data description

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

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

d. Register descriptions

Registers are described as shown below.

- Register name <Bit Symbol>

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

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

- Register name [Bit]

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

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

Revision History

Date	Revision	Comment
2019-02-18	1	First Release
2022-05-20	2	Contents Revised

Table of Contents

General precautions on the use of Toshiba MCUs

TMPM3U6FY/FW

2. Processor Core

3. Memory Map

4. Reset Operation

5. Clock/Mode control

6. Exceptions

7. DMA Controller (DMAC)

8. Input / Output Ports

9. 16-bit Timer / Event Counters (TMRB)

10. 16-bit Multi-Purpose Timer (MPT)

11. Serial Channel (SIO/UART)

12. Serial Bus Interface (I2C/SIO)

13. Synchronous Serial Port (SSP)

14. Remote control signal preprocessor(RMC)

15. Analog/Digital Converter (ADC)

16. Encoder Input Circuit (ENC)

17. Real Time Clock (RTC)

18. Power-on-Reset Circuit (POR)

19. Voltage Detection Circuit (VLTD)

20. Oscillation Frequency Detector (OFD)

21. Watchdog Timer(WDT)

22. Flash

23. ROM protection

24. Debug Interface

25. Port Section Equivalent Circuit Schematic

26. Electrical Characteristics

27. Package Dimensions

CMOS 32-Bit Microcontroller

TMPM3U6FW/FYFG
TMPM3U6FW/FYDFG

The TMPM3U6FY/FW are a 32-bit RISC microprocessor with an Arm® Cortex® -M3 microprocessor core.

Product Name	ROM (FLASH)	RAM	Package
TMPM3U6FYFG	256 Kbytes	16 Kbytes	LQFP100 (14 mm × 14 mm, 0.5 mm pitch)
TMPM3U6FWFG	128 Kbytes	12 Kbytes	
TMPM3U6FYDFG	256 Kbytes	16 Kbytes	QFP100 (14 mm × 20 mm, 0.65 mm pitch)
TMPM3U6FWDFG	128 Kbytes	12 Kbytes	

The outlines and features are as follows:

1.1 Features

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

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

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

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

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

2. On-chip program memory and data memory
   - On-chip Flash ROM:

     FY:256K bytes, FW:128K bytes
   - On-chip RAM:

     FY:16K bytes, FW:12K bytes

3. DMA controller (DMAC): 2 channel

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

4. 16-bit timer (TMRB): 8 channels
    - 16-bit interval timer mode
    - 16-bit event counter mode
    - 16-bit programmable square-wave output mode (PPG)
    - External trigger 16-bit programmable square-wave output mode (PPG)
    - Timer synchronous mode
    - Input capture function

5. Real-time clock (RTC): 1 channel
    - Clock (hour, minute and second)
    - Calendar (Month, week, date and leap year)
    - Alarm (Alarm output)
    - Alarm interruption

6. Watchdog timer (WDT): 1 channel
    - Watchdog timer (WDT) generates a reset or a non-maskable interrupt (NMI).

7. Power-on Reset Circuit (POR)

8. Voltage Detection Circuit (VLTD)

9. Oscillation Frequency Detector (OFD): 1 channel

10. Encoder input circuit (ENC): 2 channels
    - Correspond to incremental encoder (AB/ABZ)
    - Rotation direction detecting
    - Counter for absolute position detection
    - Comparator for position detection
    - Noise filter
    - 3 phase sensor input

11. 16-bit Multi Purpose Timer (MPT): 3 channels
    - Motor control (PMD: 2 channels)
    - IGBT control
    - 16-bit timer

12. General-purpose serial interface (SIO/UART): 5 channels

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

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

    Either I2C bus mode or synchronous mode can be selected.

14. Synchronous serial interface (SSP): 2 channels
    - Supports SPI/SSI/Microwire frame formats
    - Transfer speed master mode: 10 Mbps (max.), slave mode 3.3 Mbps (max.)

15. Remote control signal preprocessor (RMC): 1 channel

    Can receive up to 72 bits data at a time

16. 12/10-bit AD converter (ADC): 1 units (Analog input 18 channels)
    - Start by the internal trigger: TMRB interrupt / PMD trigger
    - 3 conversion mode (Trigger start, Software start, Constant conversion) Arbitrary AIN can be selected
    - AD Conversion Result Register (12ch)
    - AD conversion monitoring function (2ch)
    - Conversion speed 2.0 μs (@fsys = 40 MHz)

17. Interrupt source
    - Internal: 77factors. The order of precedence can be set over 7 levels (except the NMI interrupt).
    - External: 16 factors. The order of precedence can be set over 7 levels.

18. Input/ output ports: 84 pins
    - Input/output 83 pins
    - output: 1 pin

19. Standby mode
    - Standby modes: IDLE, SLEEP, STOP
    - Sub clock operation (32.768 kHz): SLOW, SLEEP

20. Clock generator (CG)
    - External Oscillator (High Frequency 10MHz) or On-chip Oscillator (9MHz)
    - External Oscillator (Low Frequency 32.768kHz)
    - On-chip PLL (4×)
    - Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8 or 1/16.

21. Endian

    Little endian

22. Debug interface

    JTAG/SWD/SWV/TRACE (DATA 2 bits)

23. Maximum operating frequency: 40 MHz

24. Operating voltage range
    - DVDD5=4.5 V to 5.5 V (All Functions)
    - DVDD5=4.0 V to 5.5 V (without 12-bit ADC conversion accuracy, AC Characteristics and Flash writing/erasing)

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

26. Package

    LQFP100 (14 mm × 14 mm, 0.5 mm pitch)
    QFP100 (14 mm × 20 mm, 0.65 mm pitch)

1.2 Block Diagram

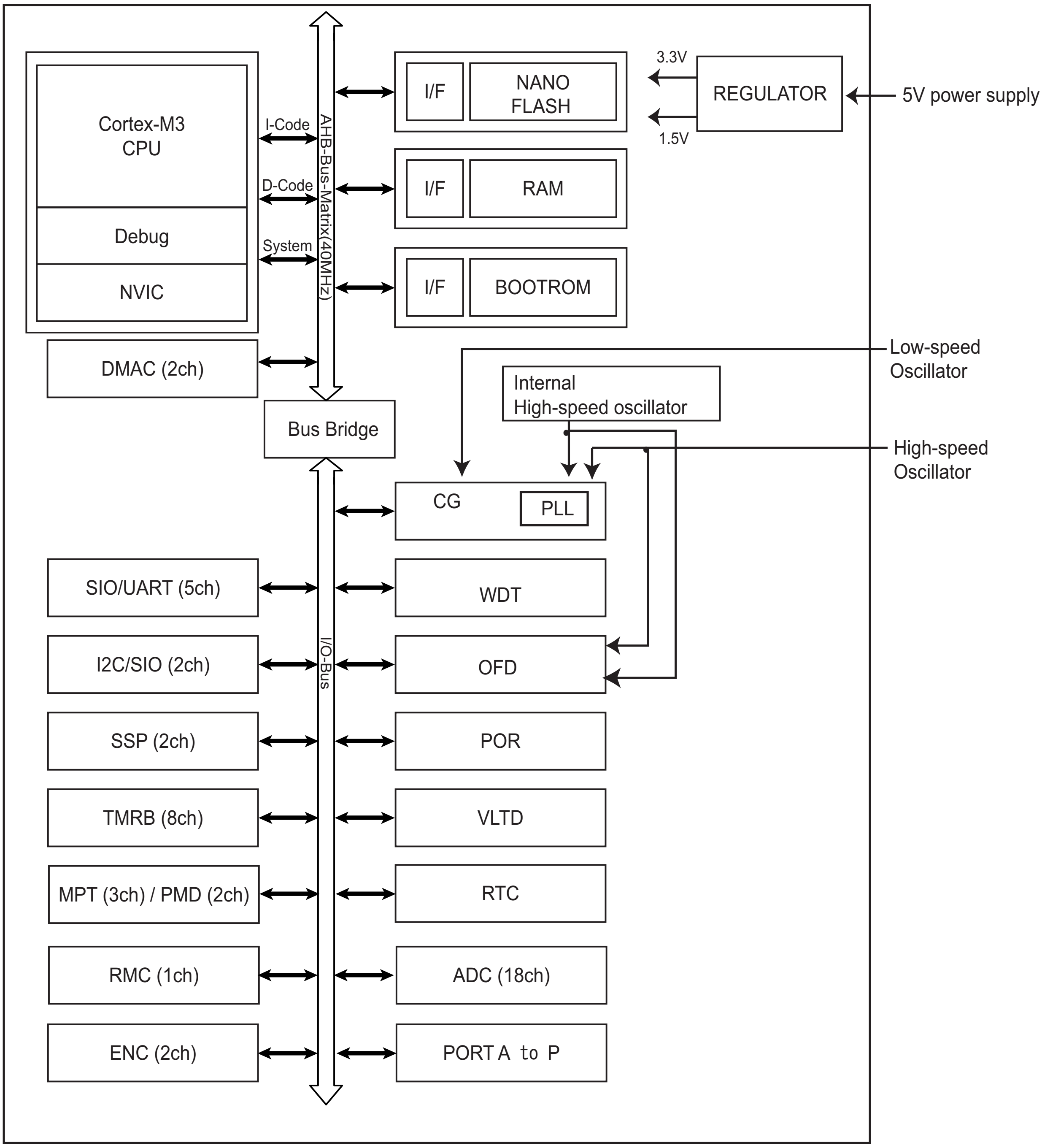


Figure 1-1 TMPM3U6FY/FW Block Diagram

1.3 Pin Layout (Top view)

Figure 1-2 shows the pin layout of TMPM3U6FY/FWFG.

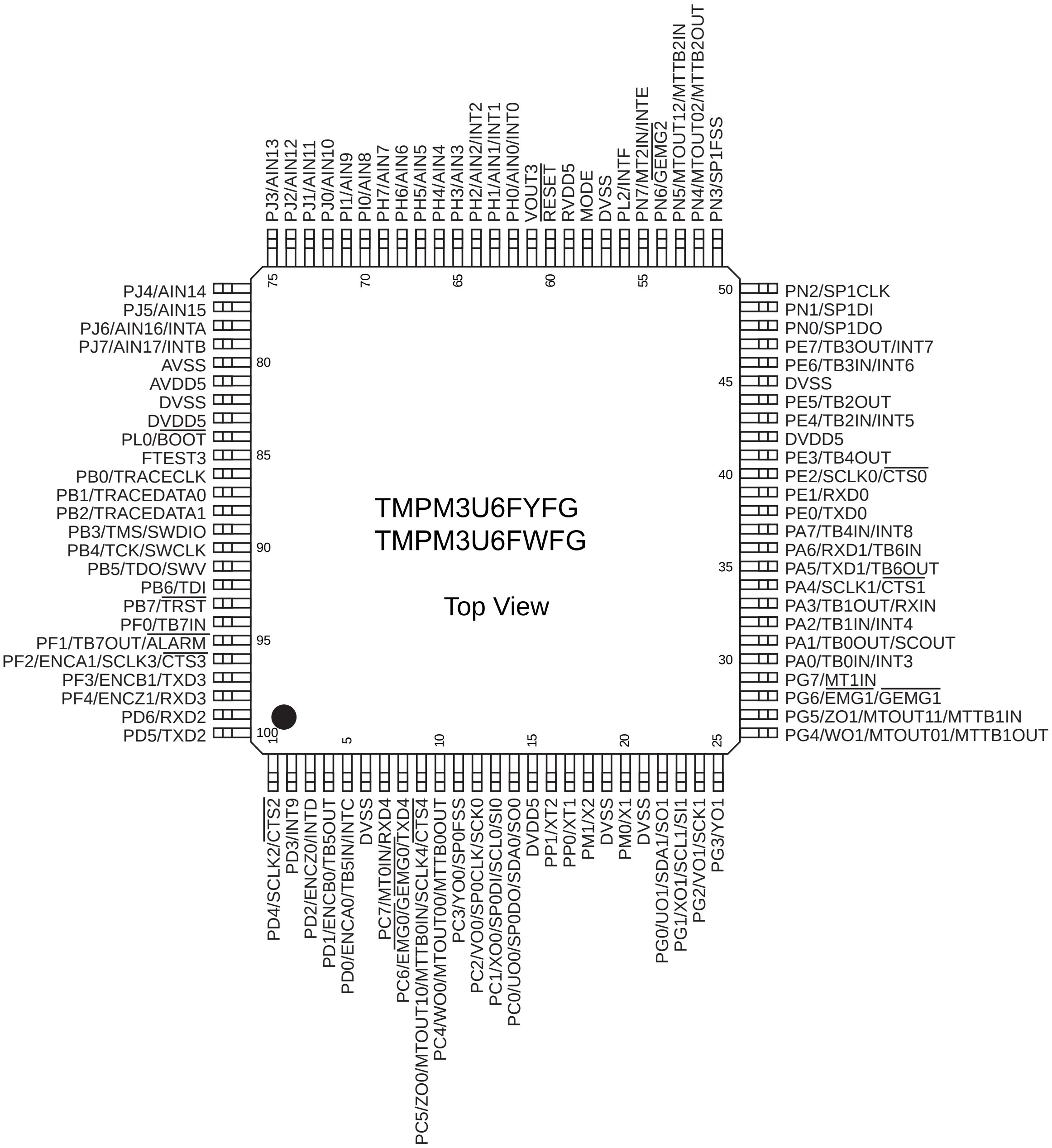


Figure 1-2 Pin Layout (LQFP100)

Figure 1-3 shows the pin layout of TMPM3U6FY/FWDFG.

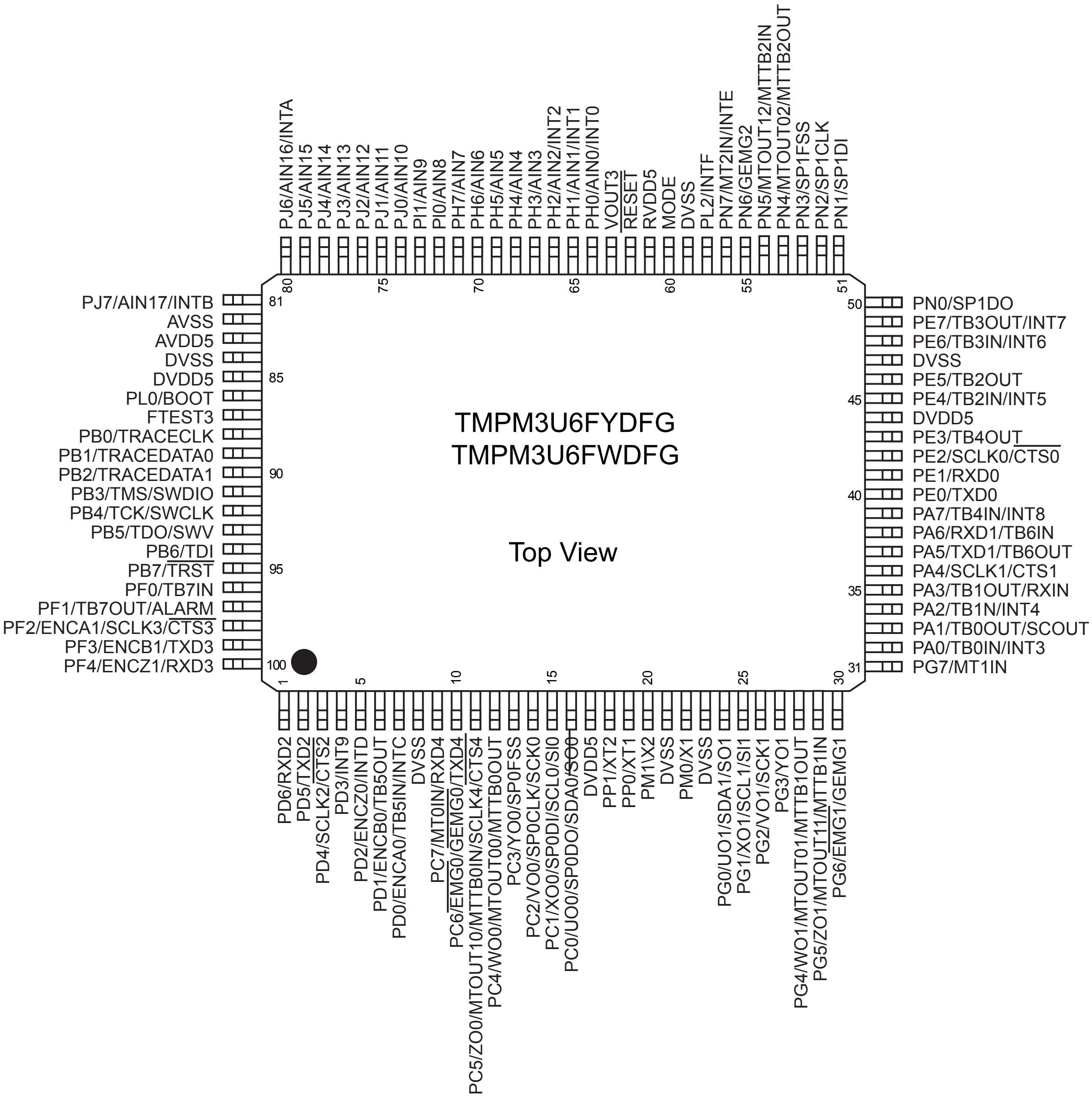


Figure 1-3 Pin Layout (QFP100)

1.4 Pin names and Functions

Table 1-1 shows the input/output pin names and functions of TMPM3U6FY/FW.

1.4.1 Sorted by Pin Number

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

Type	LQFP 100	QFP 100.	Pin Name	Input/ Output	Function
Function	1	3	PD4 SCLK2 $\overline{\text{CTS2}}$	Input/Output Input/Output Input	Input/Output port SIO clock pin SIO Handshake pin
Function	2	4	PD3 INT9	Input/Output Input	Input/Output port External interrupt pin
Function	3	5	PD2 ENCZ0 INTD	Input/Output Input Input	Input/Output port Z-phase input pin External interrupt pin
Function	4	6	PD1 ENCB0 TB5OUT	Input/Output Input Output	Input/Output port B-phase input pin Timer B output pin
Function	5	7	PD0 ENCA0 TB5IN INTC	Input/Output Input Input Input	Input/Output port A-phase input pin Inputting the timer B capture trigger External interrupt pin
Power supply	6	8	DVSS	-	GND pin
Function	7	9	PC7 MT0IN RXD4	Input/Output Input Input	Input/Output port Multi-purpose timer (IGBT mode) input pin SIO receive pin
Function	8	10	PC6 $\overline{\text{EMG0}}$ $\overline{\text{GEMG0}}$ TXD4	Input/Output Input Input Output	Input/Output port Multi-purpose timer (PMD mode) abnormal status detection input Multi-purpose timer (IGBT mode) abnormal status detection input SIO transmit pin
Function	9	11	PC5 ZO0 MTOUT10 MTTB0IN SCLK4 $\overline{\text{CTS4}}$	Input/Output Output Output Input Input/Output Input	Input/Output port Multi-purpose timer (PMD mode) Z-phase output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin SIO clock pin SIO Handshake pin
Function	10	12	PC4 WO0 MTOUT00 MTTB0OUT	Input/Output Output Output Output	Input/Output port Multi-purpose timer (PMD mode) W-phase output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin
Function	11	13	PC3 YO0 SP0FSS	Input/Output Output Input/Output	Input/Output port Multi-purpose timer (PMD mode) Y-phase output pin SSP frame/slave selection pin

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

Type	LQFP 100	QFP 100.	Pin Name	Input/ Output	Function
Function	12	14	PC2 VO0 SP0CLK SCK0	Input/Output Output Input/Output Input/Output	Input/Output port Multi-purpose timer (PMD mode) V-phase output pin SSP clock pin SIO mode clock pin
Function	13	15	PC1 XO0 SP0DI SCL0/SI0	Input/Output Output Input Input/Output	Input/Output port Multi-purpose timer (PMD mode) X-phase output pin SSP data input pin I2C mode clock pin, SIO mode receive pin
Function	14	16	PC0 UO0 SP0DO SDA0/SO0	Input/Output Output Output Input/Output	Input/Output port Multi-purpose timer (PMD mode) U-phase output pin SSP data output pin I2C mode transmit/receive pin, SIO mode transmit pin
Power supply	15	17	DVDD5	-	Power supply pin
Clock	16	18	PP1 XT2	Input/Output Output	Input/Output port Connect to a Low-speed oscillator
Clock	17	19	PP0 XT1	Input/Output Input	Input/Output port Connect to a Low-speed oscillator
Clock	18	20	PM1 X2	Input/Output Output	Input/Output port Connect to a high-speed oscillator
Power supply	19	21	DVSS	-	GND pin
Clock	20	22	PM0 X1	Input/Output Input	Input/Output port Connect to a high-speed oscillator
Power supply	21	23	DVSS	-	GND pin
Function	22	24	PG0 UO1 SDA1/SO1	Input/Output Output Input/Output	Input/Output port Multi-purpose timer (PMD mode) U-phase output pin I2C mode transmit/receive pin, SIO mode transmit pin
Function	23	25	PG1 XO1 SCL1/SI1	Input/Output Output Input/Output	Input/Output port Multi-purpose timer (PMD mode) X-phase output pin I2C mode clock pin, SIO mode receive pin
Function	24	26	PG2 VO1 SCK1	Input/Output Output Input/Output	Input/Output port Multi-purpose timer (PMD mode) V-phase output pin SIO mode clock pin
Function	25	27	PG3 YO1	Input/Output Output	Input/Output port Multi-purpose timer (PMD mode) Y-phase output pin
Function	26	28	PG4 WO1 MTOUT01 MTTB1OUT	Input/Output Output Output Output	Input/Output port Multi-purpose timer (PMD mode) W-phase output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin
Function	27	29	PG5 ZO1 MTOUT11 MTTB1IN	Input/Output Output Output Input	Input/Output port Multi-purpose timer (PMD mode) Z-phase output pin Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin

Table 1-1 Pin Names and Functions Sorted by Pin Number (3/6)

Type	LQFP 100	QFP 100.	Pin Name	Input/ Output	Function
Function	28	30	PG6 $\overline{\text{EMG1}}$ $\overline{\text{GEMG1}}$	Input/Output Input Input	Input/Output port Multi-purpose timer (PMD mode) abnormal status detection input Multi-purpose timer (IGBT mode) abnormal status detection input
Function	29	31	PG7 MT1IN	Input/Output Input	Input/Output port Multi-purpose timer (IGBT mode) input pin
Function	30	32	PA0 TB0IN INT3	Input/Output Input Input	Input/Output port Inputting the timer B capture trigger External interrupt pin
Function	31	33	PA1 TB0OUT SCOUT	Input/Output Output Output	Input/Output port Timer B output pin Internal clock output pin
Function	32	34	PA2 TB1IN INT4	Input/Output Input Input	Input/Output port Inputting the timer B capture trigger External interrupt pin
Function	33	35	PA3 TB1OUT RXIN	Input/Output Output Input	Input/Output port Timer B output pin Remote control input pin
Function	34	36	PA4 SCLK1 $\overline{\text{CTS1}}$	Input/Output Input/Output Input	Input/Output port SIO clock pin SIO Handshake pin
Function	35	37	PA5 TXD1 TB6OUT	Input/Output Output Output	Input/Output port SIO transmit pin Timer B output pin
Function	36	38	PA6 RXD1 TB6IN	Input/Output Input Input	Input/Output port SIO receive pin Inputting the timer B capture trigger
Function	37	39	PA7 TB4IN INT8	Input/Output Input Input	Input/Output port Inputting the timer B capture trigger External interrupt pin
Function	38	40	PE0 TXD0	Input/Output Output	Input/Output port SIO transmit pin
Function	39	41	PE1 RXD0	Input/Output Input	Input/Output port SIO receive pin
Function	40	42	PE2 SCLK0 $\overline{\text{CTS0}}$	Input/Output Input/Output Input	Input/Output port SIO clock pin SIO Handshake pin
Function	41	43	PE3 TB4OUT	Input/Output Output	Input/Output port Timer B output pin
Power supply	42	44	DVDD5	-	Power supply pin
Function	43	45	PE4 TB2IN INT5	Input/Output Input Input	Input/Output port Inputting the timer B capture trigger External interrupt pin

Table 1-1 Pin Names and Functions Sorted by Pin Number (4/6)

Type	LQFP 100	QFP 100.	Pin Name	Input/ Output	Function
Function	44	46	PE5 TB2OUT	Input/Output Output	Input/Output port Timer B output pin
Power supply	45	47	DVSS	-	GND pin
Function	46	48	PE6 TB3IN INT6	Input/Output Input Input	Input/Output port Inputting the timer B capture trigger External interrupt pin
Function	47	49	PE7 TB3OUT INT7	Input/Output Output Input	Input/Output port Timer B output pin External interrupt pin
Function	48	50	PN0 SP1DO	Input/Output Output	Input/Output port SSP data output pin
Function	49	51	PN1 SP1DI	Input/Output Input	Input/Output port SSP data input pin
Function	50	52	PN2 SP1CLK	Input/Output Input/Output	Input/Output port SSP clock pin
Function	51	53	PN3 SP1FSS	Input/Output Input/Output	Input/Output port SSP frame/slave selection pin
Function	52	54	PN4 MTOUT02 MTTB2OUT	Input/Output Output Output	Input/Output port Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) output pin
Function	53	55	PN5 MTOUT12 MTTB2IN	Input/Output Output Input	Input/Output port Multi-purpose timer (IGBT mode) output pin Multi-purpose timer (timer mode) input pin
Function	54	56	PN6 $\overline{\text{GEMG2}}$	Input/Output Input	Input/Output port Multi-purpose timer (IGBT mode abnormal status detection input
Function	55	57	PN7 MT2IN INTE	Input/Output Input Input	Input/Output port Multi-purpose timer (IGBT mode) input pin External interrupt pin
Function	56	58	PL2 INTF	Input/Output Input	Input/Output port External interrupt pin
Power supply	57	59	DVSS	-	GND pin
TEST	58	60	MODE	Input	Mode pin (note) MODE pin must be connected to GND.
Power supply	59	61	RVDD5	-	Power supply pin
Reset	60	62	$\overline{\text{RESET}}$	Input	Reset input pin
Power supply	61	63	VOUT3	-	The pin which is connected to the capacitor to stabilize an internal power supply voltage. Note: Connect a capacitor (3.3 to 4.7μF) between capacitor connect pins and DVSS.
Function	62	64	PH0 AIN0 INT0	Input/Output Input Input	Input/Output port Analog input External interrupt pin

Table 1-1 Pin Names and Functions Sorted by Pin Number (5/6)

Type	LQFP 100	QFP 100.	Pin Name	Input/ Output	Function
Function	63	65	PH1 AIN1 INT1	Input/Output Input Input	Input/Output port Analog input External interrupt pin
Function	64	66	PH2 AIN2 INT2	Input/Output Input Input	Input/Output port Analog input External interrupt pin
Function	65	67	PH3 AIN3	Input/Output Input	Input/Output port Analog input
Function	66	68	PH4 AIN4	Input/Output Input	Input/Output port Analog input
Function	67	69	PH5 AIN5	Input/Output Input	Input/Output port Analog input
Function	68	70	PH6 AIN6	Input/Output Input	Input/Output port Analog input
Function	69	71	PH7 AIN7	Input/Output Input	Input/Output port Analog input
Function	70	72	PI0 AIN8	Input/Output Input	Input/Output port Analog input
Function	71	73	PI1 AIN9	Input/Output Input	Input/Output port Analog input
Function	72	74	PJ0 AIN10	Input/Output Input	Input/Output port Analog input
Function	73	75	PJ1 AIN11	Input/Output Input	Input/Output port Analog input
Function	74	76	PJ2 AIN12	Input/Output Input	Input/Output port Analog input
Function	75	77	PJ3 AIN13	Input/Output Input	Input port Analog input
Function	76	78	PJ4 AIN14	Input/Output Input	Input/Output port Analog input
Function	77	79	PJ5 AIN15	Input/Output Input	Input/Output port Analog input
Function	78	80	PJ6 AIN16 INTA	Input/Output Input Input	Input/Output port Analog input External interrupt pin
Function	79	81	PJ7 AIN17 INTB	Input/Output Input Input	Input port Analog input External interrupt pin
Power supply	80	82	AVSS	-	AD converter GND pin (note) AVSS must be connected to GND even if the AD converter is not used.
Power supply	81	83	AVDD5	-	Supplying the AD converter with a power supply. (note) AVDD5 must be connected to power supply even if AD converter is not used.

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

Type	LQFP 100	QFP 100.	Pin Name	Input/ Output	Function
Power supply	82	84	DVSS	-	GND pin
Power supply	83	85	DVDD5	-	Power supply pin
Function/ Control	84	86	PL0 $\overline{\text{BOOT}}$	Output Input	Output port $\overline{\text{BOOT}}$ mode pin
TEST	85	87	FTEST3	-	TEST pin (note) TEST pin must be left OPEN.
Function/ Debug	86	88	PB0 TRACECLK	Input/Output Output	Input/Output port Debug pin
Function/ Debug	87	89	PB1 TRACEDATA0	Input/Output Output	Input/Output port Debug pin
Function/ Debug	88	90	PB2 TRACEDATA1	Input/Output Output	Input/Output port Debug pin
Function/ Debug	89	91	PB3 TMS/SWDIO	Input/Output Input/Output	Input/Output port Debug pin
Function/ Debug	90	92	PB4 TCK/SWCLK	Input/Output Input/Output	Input/Output port Debug pin
Function/ Debug	91	93	PB5 TDO/SWV	Input/Output Output	Input/Output port Debug pin
Function/ Debug	92	94	PB6 TDI	Input/Output Input	Input/Output port Debug pin
Function/ Debug	93	95	PB7 $\overline{\text{TRST}}$	Input/Output Input	Input/Output port Debug pin
Function	94	96	PF0 TB7IN	Input/Output Input	Input/Output port Inputting the timer B capture trigger
Function	95	97	PF1 TB7OUT $\overline{\text{ALARM}}$	Input/Output Output Output	Input/Output port Timer B output pin Alarm output pin
Function	96	98	PF2 ENCA1 SCLK3 $\overline{\text{CTS3}}$	Input/Output Input Input/Output Input	Input/Output port A-phase input pin SIO clock pin SIO Handshake pin
Function	97	99	PF3 ENCB1 TXD3	Input/Output Input Output	Input/Output port B-phase input pin SIO transmit pin
Function	98	100	PF4 ENCZ1 RXD3	Input/Output Input Input	Input/Output port Z-phase input pin SIO receive pin
Function	99	1	PD6 RXD2	Input/Output Input	Input/Output port SIO receive pin
Function	100	2	PD5 TXD2	Input/Output Output	Input/Output port SIO transmit pin

1.5 Power Supplies and Power Supply Pins

Table 1-2 Pin Numbers and Power Supplies

Power supply	Voltage range	LQFP100	QFP100	Pin name
RVDD5	4.0 to 5.5V (Internal:1.5V for oscillator pin)	59	61	−
DVDD5		15,42,83	17,44,85	PA,PB,PC,PD,PE,PF,PG,PL,PN PM,PP,X1,X2,XT1,XT2 $\overline{\text{RESET}}$,MODE,FTEST3
AVDD5		81	83	PH,PI,PJ
VOUT3	2.7 to 3.6V	61	63	−
DVSS	GND	6,19,21,45,57,82	8,21,23,47,59,84	−
AVSS		80	82	−

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 TMPM3U6FY/FW.

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
TMPM3U6FY/FWFG TMPM3U6FY/FWDFG	r2p0

2.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r2p0 are ETM™, MPU and WIC. The following table shows the configurable options in the TMPM3U6FY/FW.

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

2.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

2.3.1 Number of Interrupt Inputs

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

TMPM3U6FY/FW has 93 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, 0x03 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.

TMPM3U6FY/FW 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.

TMPM3U6FY/FW provides the same operation when SYSRESETREQ signal are output.

Note:The reset operation by <SYSRESETREQ> can not used while in SLOW mode.

2.3.5 LOCKUP

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

TMPM3U6FY/FW does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable 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, TMPM3U6FY/FW are 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.

TMPM3U6FY/FW 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.

TMPM3U6FY/FW 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 TMPM3U6FY/FW does not use this function.

3. Memory Map

3.1 Memory Map

The memory maps for TMPM3U6FY/FW are based on the Arm Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM3U6FY/FW 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 TMPM3U6FY/FW Memory Map

Figure 3-1 shows the memory map of TMPM3U6FY/FW.

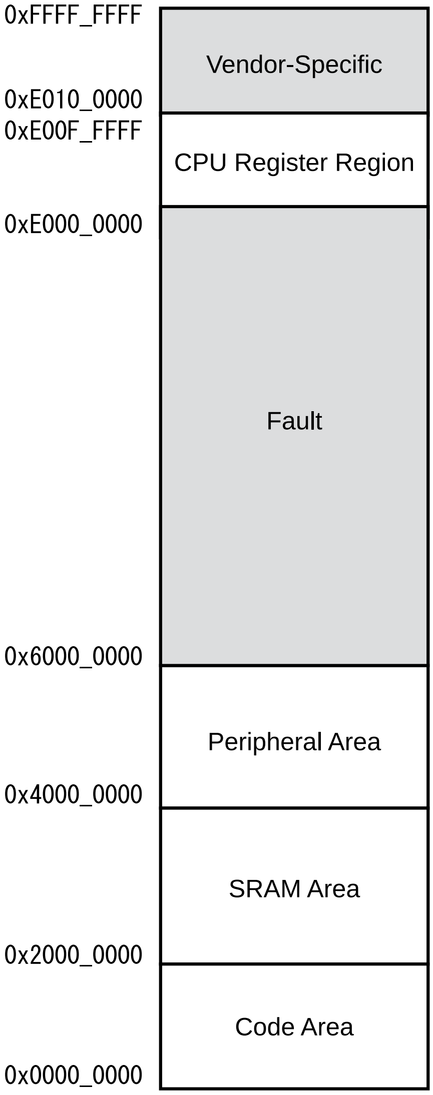


Figure 3-1 Memory Map

3.1.2 Connection Table

3.1.2.1 Code area / SRAM area

(1) Singe Chip Mode

Product name	Start Address	Slave	Master	DMAC	Core S-Bus	Core D-Bus	Core I-Bus
				S0	S1	S2	S3
TMPM3U6FW	0x0000_0000	Flash ROM	M0	Fault	Fault	o	o
	0x0002_0000	Reserved				Reserved	Reserved
	0x0004_0000	Fault	-	Fault	Fault	Fault	Fault
	0x2000_0000	RAM	M1	o	o	Fault	Fault
	0x2000_3000	Reserved		Reserved	Reserved		
	0x2000_4000	Fault	-	Fault	Fault	Fault	Fault
	0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
	0x2206_0000	Reserved			Reserved		
	0x2208_0000	Fault	-	Fault	Fault	Fault	Fault
TMPM3U6FY	0x0000_0000	Flash ROM	M0	Fault	Fault	o	o
	0x0004_0000	Fault	-	Fault	Fault	Fault	Fault
	0x2000_0000	RAM	M1	o	o	Fault	Fault
	0x2000_4000	Fault	-	Fault	Fault	Fault	Fault
	0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
	0x2208_000	Fault	-	Fault	Fault	Fault	Fault

Note: Do not access the reserved area.

(2) Single BOOT Mode

Product name	Start Address	Slave	Master	DMAC	Core S-Bus	Core D-Bus	Core I-Bus
				S0	S1	S2	S3
TMPM3U6FW	0x0000_0000	Boot ROM	M2	Fault	Fault	o	o
	0x0000_1000	Fault	-	Fault	Fault	Fault	Fault
	0x2000_0000	RAM	M1	o	o	Fault	Fault
	0x2000_3000	Reserved		Reserved	Reserved		
	0x2000_4000	Fault	-	Fault	Fault	Fault	Fault
	0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
	0x2206_0000	Reserved			Reserved		
	0x2208_0000	Fault	-	Fault	Fault	Fault	Fault
	0x3F7F_F000	Reserved	-	Fault	Reserved	Fault	Fault
	0x3F80_0000	Flash ROM(Mirror)	-	Fault	o	Fault	Fault
	0x3F82_0000	Reserved			Reserved		
	0x3F84_0000	Fault	-	Fault	Fault	Fault	Fault
TMPM3U6FY	0x0000_0000	Boot ROM	M2	Fault	Fault	o	o
	0x0000_1000	Fault	-	Fault	Fault	Fault	Fault
	0x2000_0000	RAM	M1	o	o	Fault	Fault
	0x200_4000	Fault	-	Fault	Fault	Fault	Fault
	0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
	0x2208_0000	Fault	-	Fault	Fault	Fault	Fault
	0x3F7F_F000	Reserved	-	Fault	Reserved	Fault	Fault
	0x3F80_0000	Flash ROM(Mirror)	-	Fault	o	Fault	Fault
	0x3F84_0000	Fault	-	Fault	Fault	Fault	Fault

Note: Do not access the reserved area.

3.1.2.2 Peripheral area / External bus area

Start Address	Master		DMAC	Core S-Bus	Core D-Bus	Core I-Bus
	Slave		S0	S1	S2	S3
0x4000_0000	PORT	M3	o	o	Fault	Fault
0x4001_0000	TMRB		o	o	Fault	Fault
0x4001_0400	ENC		o	o	Fault	Fault
0x4002_0000	SBI		o	o	Fault	Fault
0x4002_0080	SIO/UART		o	o	Fault	Fault
0x4003_0000	ADC		o	o	Fault	Fault
0x4004_0000	WDT		o	o	Fault	Fault
0x4004_0100	RTC		o	o	Fault	Fault
0x4004_0200	CG		o	o	Fault	Fault
0x4004_0400	RMC		o	o	Fault	Fault
0x4004_0800	OFD		o	o	Fault	Fault
0x4004_0900	VLTD		o	o	Fault	Fault
0x4005_0400	MPT(PMD)		o	o	Fault	Fault
0x4005_0800	MPT(TMRB/IGBT)		o	o	Fault	Fault
0x4008_0000	DMAC	M4	Fault	o	Fault	Fault
0x4009_0000	Fault	-	Fault	Fault	Fault	Fault
0x400C_0000	SSP	M5	o	o	Fault	Fault
0x400C_4000	Fault	-	Fault	Fault	Fault	Fault
0x41FF_F000	Flash(SFR)	M3	o	o	Fault	Fault
0x4200_0000	Bit Band Alias	-	Fault	o	Fault	Fault
0x4320_0000	Fault	-	Fault	Fault	Fault	Fault
0x4380_0000	Bit Band Alias	-	Fault	o	Fault	Fault
0x4388_0000	Fault	-	Fault	Fault	Fault	Fault

3.2 Table of the Peripheral base address

Do not access the area except control registers. Reading the detail of the control registers, please refer to the each peripheral function’s section.

Peripheral function		Base address
PORT	Port A	0x4000_0000
	Port B	0x4000_0040
	Port C	0x4000_0080
	Port D	0x4000_00C0
	Port E	0x4000_0100
	Port F	0x4000_0140
	Port G	0x4000_0180
	Port H	0x4000_01C0
	Port I	0x4000_0200
	Port J	0x4000_0240
	Port L	0x4000_02C0
	Port M	0x4000_0300
	Port N	0x4000_0340
	Port P	0x4000_0380
16-bit timer / event counter (TMRB)	ch0	0x4001_0000
	ch1	0x4001_0040
	ch2	0x4001_0080
	ch3	0x4001_00C0
	ch4	0x4001_0100
	ch5	0x4001_0140
	ch6	0x4001_0180
	ch7	0x4001_01C0
Encorder input circuit (ENC)	ch0	0x4001_0400
	ch1	0x4001_0500
Serial bus interface (I2C/SIO)	ch0	0x4002_0000
	ch1	0x4002_0020
Serial channel (SIO/UART)	ch0	0x4002_0080
	ch1	0x4002_00C0
	ch2	0x4002_0100
	ch3	0x4002_0140
	ch4	0x4002_0180
Analog / digital converter (ADC)		0x4003_0000
Watch dog timer (WDT)		0x4004_0000
Real time clock (RTC)		0x4004_0100
Clock / mode control (CG)		0x4004_0200
Remote control signal preprocessor (RMC)		0x4004_0400
Oscillation frequency detector (OFD)		0x4004_0800
Voltage detection circuit (VLTD)		0x4004_0900
16-bit multi-purpose timer (PMD)	ch0	0x4005_0400
	ch1	0x4005_0480
16-bit multi-purpose timer (TMRB/IGBT)	ch0	0x4005_0800
	ch1	0x4005_0880
	ch2	0x4005_0900
DMA controller (DMAC)		0x4008_0000
Synchronous Serial Port (SSP)	ch0	0x400C_0000
	ch1	0x400C_1000

Peripheral function	Base address
Flash control (Flash SFR)	0x41FF_F000

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

Note 2: In the SLOW mode, do not use a reset by <SYSRESETREQ>.

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. TMPM3U6FY/FW has a function to insert a stable time automat-ically.

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 0.8ms internal reset signal is released.

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

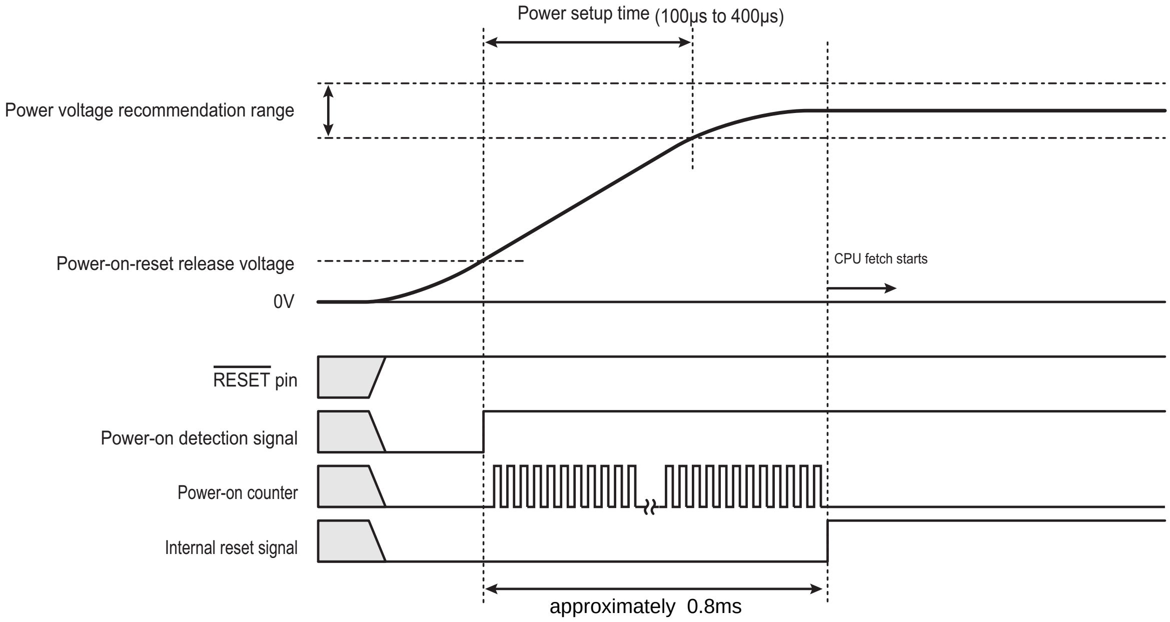

Power setup time (100μs to 400μs)
Power voltage recommendation range
Power-on-reset release voltage
0V
CPU fetch starts
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal
approximately 0.8ms

Figure 4-1 Reset Operation by Power-on Circuit

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

4.1.2 Reset by RESET pin

Internal reset signal is released approximately 0.4ms after RESET pin becomes "High". However if RESET pin is set to "High" within 400µs after power-on reset signal becomes "High", the reset process will be the same as the power-on described in 4.1.1.

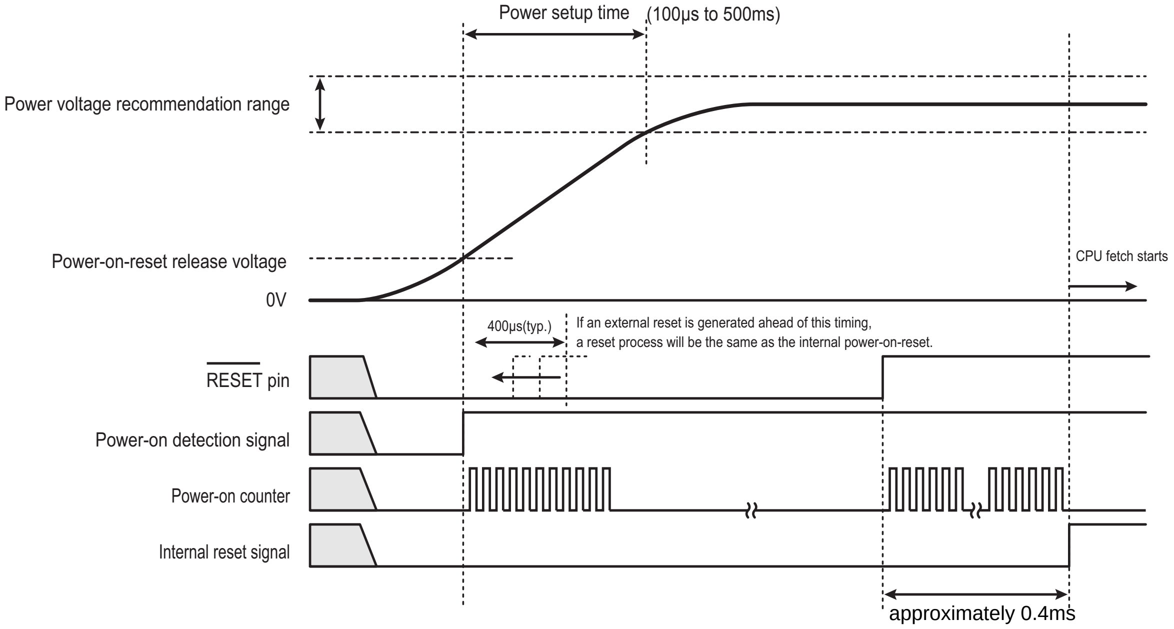

Power setup time
(100µs to 500ms)
Power voltage recommendation range
Power-on-reset release voltage
0V
CPU fetch starts
400µs(typ.)
If an external reset is generated ahead of this timing,
a reset process will be the same as the internal power-on-reset.
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal
approximately 0.4ms

Figure 4-2 Reset Operation by RESET pin

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

4.2 Warm-up

4.2.1 Reset Duration

To do reset TMPM3U6FY/FW, the following condition is required; power supply voltage is in the operational range; $\overline{\text{RESET}}$ pin is kept "Low" at least for 12 system clocks by internal high frequency oscillator. Approx-imately 0.4ms 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 Outline

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

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

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

5.2 Registers

5.2.1 Register List

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

Base Address = 0x4004_0200

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

5.2.2 CGSYSCR (System control register)

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

Bit	Bit Symbol	Type	Function
31-24	−	R	Read as "0".
23	−	R/W	Write "0".
22-21	−	R	Read as "0".
20	FCSTOP	R/W	ADC clock 0: Active 1: Stop Enables to stop providing AD converter clock. AD converter clock is provided after reset. Confirming that AD converter is stopped or finished in advance is required when setting "1" (stop).
19-18	−	R	Read as "0".
17-16	SCOSEL[1:0]	R/W	SCOUT out 00: fs 01: fsys/2 10: fsys 11: φT0 Enables to output the specified clock from SCOUT pin.
15-14	−	R	Read as "0".
13-12	FPSEL[1:0]	R/W	φT0 source clock 00: The clock divided fgear by a prescaler 01: The clock divided fc by a prescaler 10: fsys 11: fsys Specifies the source clock to φT0. In the SLOW mode, must be set "10" or "11".
11	−	R	Read as "0".
10-8	PRCK[2:0]	R/W	Prescaler clock 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved Specifies the prescaler clock to peripheral functions.
7-3	−	R	Read as "0".
2-0	GEAR[2:0]	R/W	High-speed clock (fc) gear 000: fc 100: fc/2 001: Reserved 101: fc/4 010: Reserved 110: fc/8 011: Reserved 111: fc/16

5.2.3 CGOSCCR (Oscillation control register)

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

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

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

Note 1: Refer to Section "5.3.4 Warm-up function" about the Warm-up setup.

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

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

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

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	-	-	-	-	-	-	RXTEN	RXEN
After reset	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	STBY		
After reset	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-20	−	R	Read as "0".
19-17	−	R/W	Write "0".
16	DRVE	R/W	Controls the port in STOP mode 0: Not drive ports 1: Drive ports
15-10	−	R	Read as "0".
9	RXTEN	R/W	Low-speed oscillator operation after releasing STOP mode. 0: Stop 1: Oscillation
8	RXEN	R/W	High-speed oscillator operation after releasing STOP mode. 0: Stop 1: Oscillation
7-3	−	R	Read as "0".
2-0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP 010: SLEEP 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved

Note: Access to the "Reserved" is prohibited.

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-12	−	R/W	Write "0111"
11	−	R	Read as "0".
10-8	−	R/W	Write "010"
7-1	−	R/W	Write "0001111"
0	PLLSEL	R/W	Use of PLL 0: fosc use 1: f_{PLL} use Specifies use or disuse of the clock multiplied by the PLL. "fosc (internal high-speed oscillator)" is automatically set after reset.Resetting is required when using the PLL.

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

5.2.6 CGCKSEL (System clock selection register)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	SYSCK	R/W	Selects system clock 0: high-speed 1: low-speed Specifies system clock. When change value of <SYSCK>, oscillation must stable High-speed oscillator (f_{EHOSC} or f_{IHOSC}) and Low-speed oscillator. According to the used oscillator, corresponding CGOSCCR<XEN1>, <XEN2> or <XTEN> must be set to "1" in advance.
0	SYSCKFLG	R	System clock status 0: high-speed 1: low-speed Shows the status of the system clock. When switching the oscillator with <SYSCK>, generates time lag to complete. If the read value from <SYSCKFLG> is the same as the value specified in <SYSCK>, the switching has been completed.

5.3 Clock control

5.3.1 Clock Type

Each clock is defined as follows:

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

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

fs : Clock generated by external low-speed oscillator.

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

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

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

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

fsys : Clock specified by CGCKSEL<SYSCK>

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

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

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

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

5.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

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

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

$fc = f_{IHOSC}$

$fsys = fc = f_{IHOSC}$

$fperiph = fc = f_{IHOSC}$

$φT0 = fperiph = f_{IHOSC}$

5.3.3 Clock system Diagram

Figure 5-1 shows the clock system diagram.

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

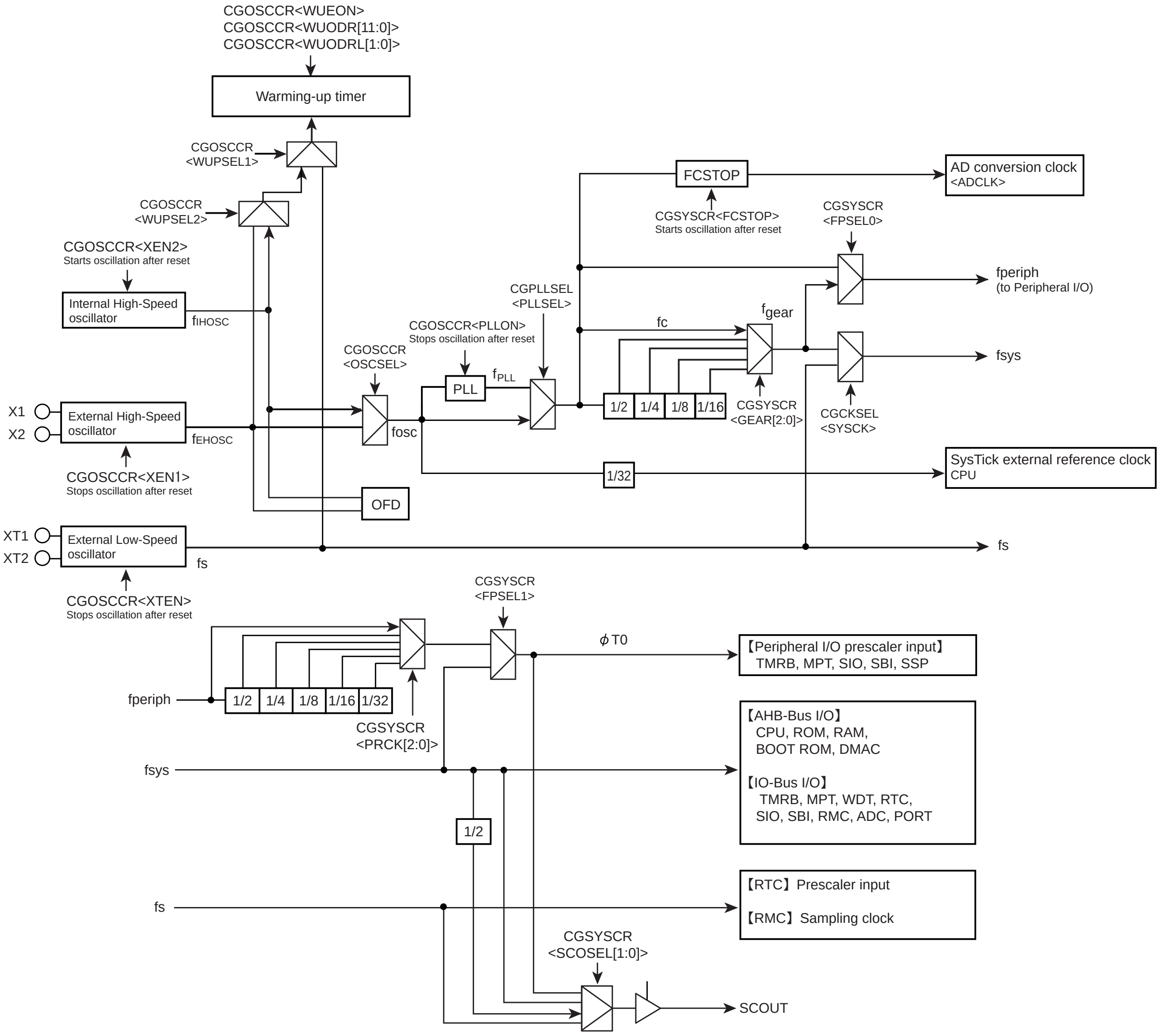


Figure 5-1 Clock Block Diagram

5.3.4 Warm-up function

The warm-up function secures the stability time for the oscillator of fs and the PLL with the warm-up timer when releasing STOP mode. Refer to "5.3.4 Warm-up function" for a detail.

How to configure the warm-up function.

1. Specify the count up clock

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

2. Specify the warm-up counter value

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

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

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

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

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

$$\frac{\text{warm-up time to set}}{\text{input frequency cycle (s)}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40{,}000\text{ cycle} = \text{0x9C40}$$

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

3. confirm the start and completion of warm-up

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

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

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

The example of warm-up function setup.

Table 5-1 The example of warm-up setting (When an external high-speed oscillator is selected)

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

5.3.5 Clock Multiplication Circuit (PLL)

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

5.3.5.1 How to configure the PLL function

The PLL is disabled after reset.

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

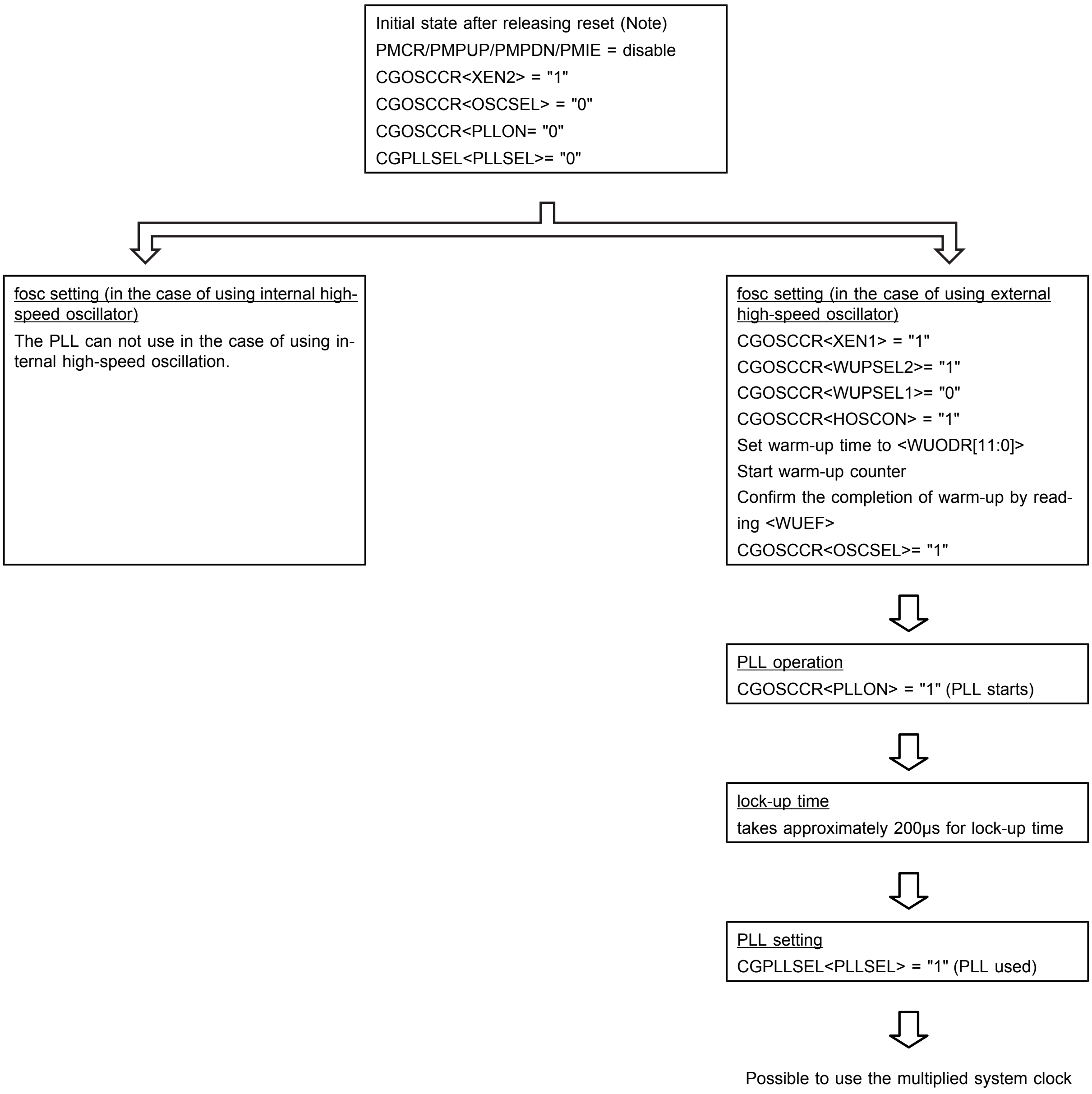
The PLL requires a certain amount of time to be stabilized, which should be secured using the warm-up function or other methods.

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

5.3.5.2 The sequence of PLL setting

The sequence of PLL setting is shown below.

The sequence of PLL setting



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

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

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

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

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

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

Possible to use the multiplied system clock

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

5.3.6 System clock

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

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

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

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

Table 5-2 The range of an operation frequency when a clock gear and PLL are used
(Unit : MHz, "-" : Reserved, "*" : Don’t care)

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

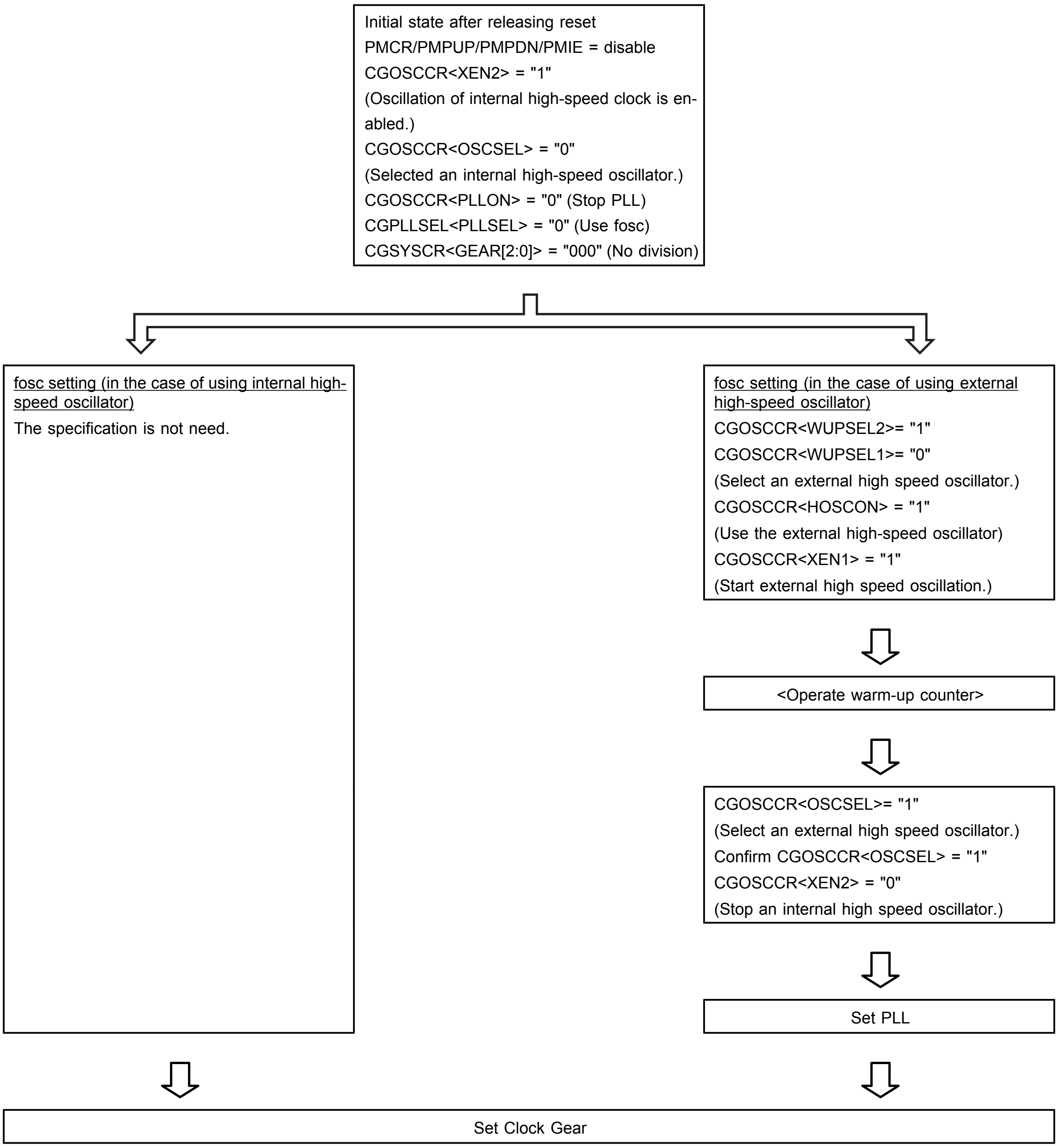
↑ Initial value after reset

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

5.3.6.1 The sequence of System clock setting

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

The sequence of PLL setting



Initial state after releasing reset
PMCR/PMPUP/PMPDN/PMIE = disable
CGOSCCR<XEN2> = "1"
(Oscillation of internal high-speed clock is enabled.)
CGOSCCR<OSCSEL> = "0"
(Selected an internal high-speed oscillator.)
CGOSCCR<PLLON> = "0" (Stop PLL)
CGPLLSEL<PLLSEL> = "0" (Use fosc)
CGSYSCR<GEAR[2:0]> = "000" (No division)

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

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

<Operate warm-up counter>

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

Set PLL

Set Clock Gear

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[1:0]> can be divided according to the setting in the CGSYSCR<PRCK[2:0]>. After the TMPM3U6FY/FW are reset, fperiph/1 is selected as φT0.

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

5.3.8 System Clock Pin Output Function

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

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

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

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

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

Table 5-3 SCOUT Output Status in Each Mode

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

Note: When TMPM3U6FY/FW are change to the STOP mode, please set SCSTBYCR<PTKEEP> to "1" first and hold the state of a port.

5.4 Modes and Mode Transitions

5.4.1 Mode Transitions

The NORMAL mode and the SLOW mode use the high-speed and low-speed clocks for the system clock respectively.

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

When the low-speed clock is not used, the SLOW and SLEEP modes cannot be used.

Figure 5-2 shows a mode transition diagram.

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

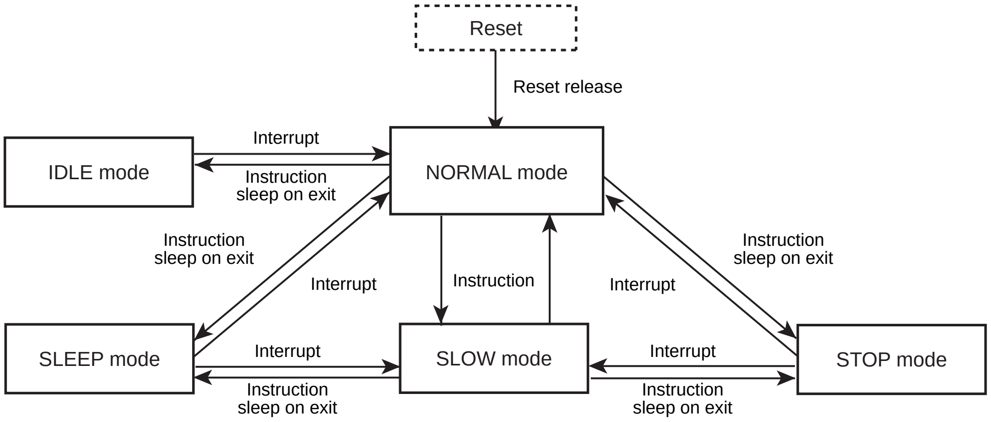


Figure 5-2 Mode Transition Diagram

5.5 Operation mode

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

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. The low-speed clock can also be oscillated.

5.5.2 SLOW mode

This mode is to operate the CPU core and the peripheral hardware by using the low-speed clock with high-speed clock stopped. The SLOW mode reduces power consumption compared to the NORMAL mode.

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

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

5.6 Low Power Consumption Modes

The TMPM3U6FY/FW has the low power consumption modes: IDLE, SLEEP and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction. In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

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

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

The features of IDLE, SLEEP, STOP mode are described as follows.

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)
- 16-bit multi purpose timer counter (MPT : except PMD function)
- Serial channel (SIO/UART)
- Serial bus interface (I2C/SIO)
- Analog Digital converter (ADC)
- Watch dog timer (WDT)

Note: WDT should be stopped before entering IDLE mode.

5.6.2 SLEEP mode

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

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

5.6.3 STOP mode

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

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

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

Note: Warm-up is need at the time of a return. It is necessary to set warm-up time in the mode (NORMAL mode or SLOW mode) before entering STOP mode. Please refer to "5.6.8 Clock Operations in Mode Transition" about warm-up time.

Table 5-4 Pin States in the STOP mode

Function	Pin Name	I/O	<DRVE> = 0	<DRVE> = 1
Oscillator	X1, XT1	Input	×	×
	X2, XT2	Output	"High" level output.	"High" level output.
PORT	Px	Input	×	Depends on PxIE[m]
		Output	×	Depends on PxCR[m]
Debug	TMS/SWDIO TDO/SWV	Input	Depends on PxIE[m]	
		Output	Depends on PxCR[m] and enable when data is valid	
Interrupt	INT	Input	Depends on PxIE[m]	
SSP	SPCLK, SPFSS, SPDO	Output	×	Depends on PxCR[m] and enable when data is valid
MPT (PMD mode)	$\overline{\text{EMG}}$	Input	×	Depends on PxIE[m]
	UO, VO, WO XO, YO, ZO	Output	Depend on PxCR[m] and enable when data is valid	
MPT (IGBT mode)	$\overline{\text{GEMG}}$, MTIN	Input	×	Depends on PxIE[m]
	MTOUT0 MTOUT1	Output	Depend on PxCR[m] and enable when data is valid	
except above		Input	×	Depends on PxIE[m]
		Output	×	Depends on PxCR[m]

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

Note: x: port number / m: corresponding bit

5.6.4 Low power Consumption Mode Setting

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

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

Table 5-5 Low power consumption mode setting

Mode	CGSTBYCR <STBY[2:0]>
STOP	001
SLEEP	010
IDLE	011

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

5.6.5 Operational Status in Each Mode

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

Table 5-6 Operational Status in Each Mode

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

o : Operation is available when in the target mode.

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

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

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

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

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

5.6.6 Releasing the Low Power Consumption Mode

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

Details are shown in Table 5-7.

Table 5-7 Release Source in Each Mode

Low power consumption mode			IDLE	SLEEP	STOP
Release source	Interrupt	INT0 to F (Note1)	o	o	o
		INTRTC	o	o	×
		INTRMCRX	o	o	×
		INTSSP0 to 1	×	×	×
		INTSBI0 to 1	o	×	×
		INTRX0 to 4, INTTX0 to 4	o	×	×
		INTADPD0 to 1 /INTADCP0 to 1	o	×	×
		INTADTMR /INTADSFT	o	×	×
		INTPMD0 to 1, INTEMG0 to 1	o	×	×
		INTMTTB00 to 20 / 01 to 21	o	×	×
		INTMTCAP00 to 20 / 01 to 21	o	×	×
		INTMTEMG0 to 2	o	×	×
		INTTB00 to 70 / 01 to 71	o	×	×
		INTCAP00 to 70 / 01 to 71	o	×	×
		INTENC0 to 1	o	×	×
		INTDMACERR, INTDMACTC	o	×	×
	SysTick Interrupt		o	×	×
	Non-Maskable Interrupt (INTWDT)		o	×	×
	Non-Maskable Interrupt (INTVLTD)		o	o	o
	Reset (WDT)		o	×	×
	Reset (POR)		o	o	o
	Reset (OFD)		o	×	×
	Reset ($\overline{\text{RESET}}$ pin)		o	o	o

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

Note 1: When releasing from IDLE, SLEEP, STOP mode by interrupting level mode, hold the level until the interrupt handling starts. If the level is changed before that, the correct interrupt handling can not be started.

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

- Release by interrupt request

  To release the low power consumption mode by an interrupt, the interrupt is set to detect interrupt request before entering the low power consumption mode.

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

- Release by SysTick interrupt

  SysTick interrupt can only be used in the IDLE mode.

- Release by Non-Maskable Interrupt (NMI)

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

  INTWDT and INTVLTD can be used only in the IDLE mode.

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

- Release by reset

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

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

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

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

Refer to "Interrupts" for details.

5.6.7 Warm-up

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

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

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

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

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

In regard to warm-up time, please refer to "5.6.8 Clock Operations in Mode Transition".

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

Table 5-8 Warm-up setting in mode transition

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

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

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

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

5.6.8 Clock Operations in Mode Transition

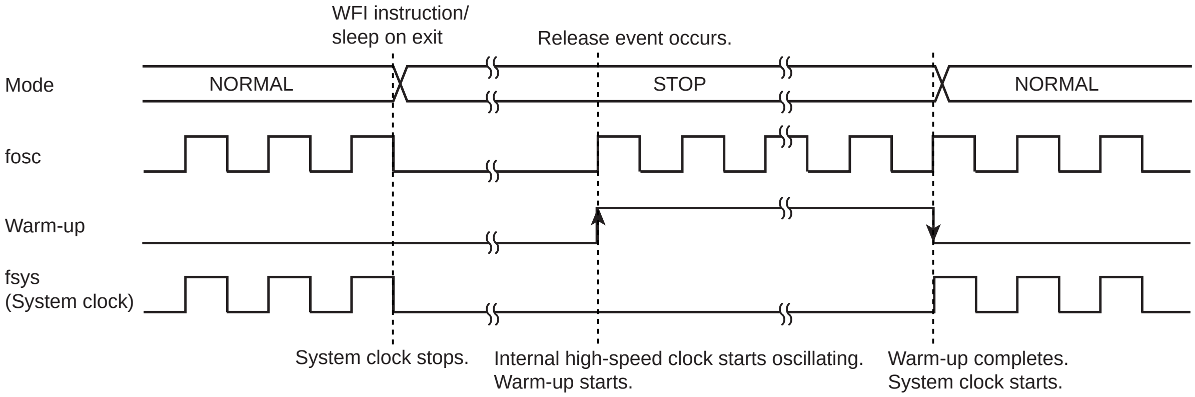
The clock operations in mode transition are described as follows.

5.6.8.1 Transition of operation modes: NORMAL → STOP → NORMAL

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

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

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

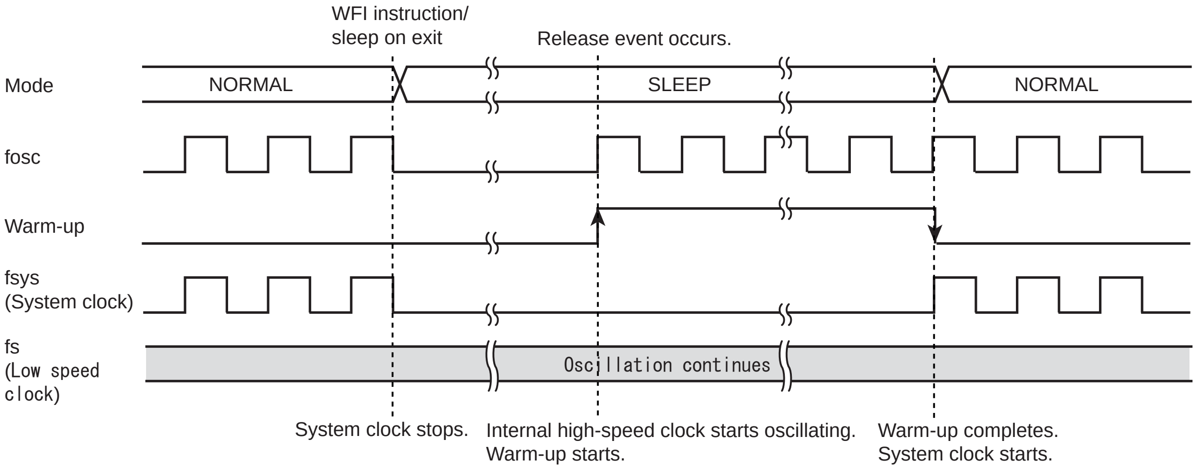


5.6.8.2 Transition of operation modes: NORMAL → SLEEP → NORMAL

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

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

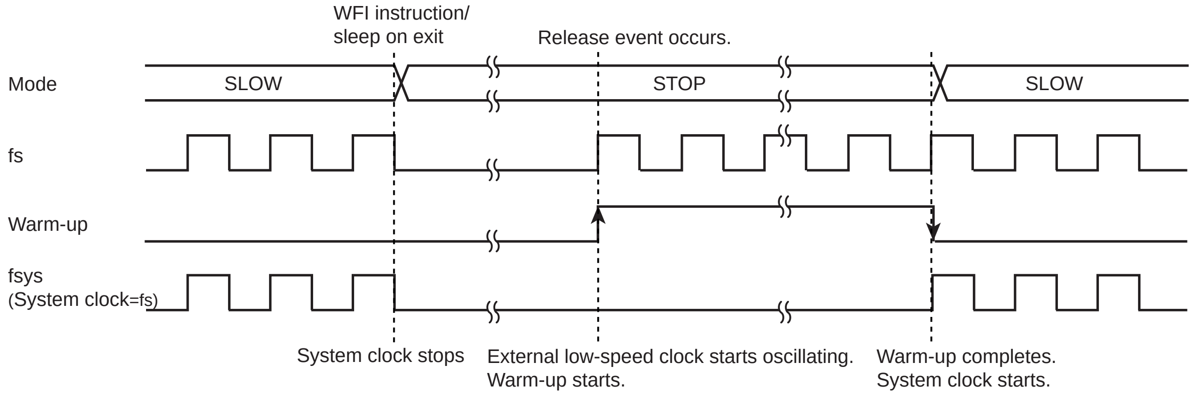
Returning to the NORMAL mode by reset does not induce the automatic warm-up. The reset signal as same as a cold reset should be input.



5.6.8.3 Transition of operation modes: SLOW → STOP → SLOW

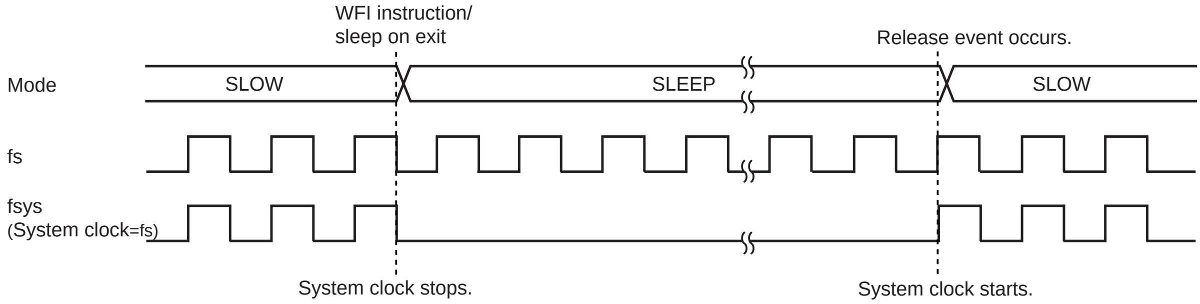
When returning to the SLOW mode from the STOP mode, the warm-up is activated automatically.

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



5.6.8.4 Transition of operation modes: SLOW → SLEEP → SLOW

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



5.6.9 Precaution on Transition to the Low-power Consumption Mode

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

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

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

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

b. Case when the interrupts are enabled

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

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

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

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

6. 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.

6.1 Overview

An exception causes the CPU to stop the currently executing process and handle another process.

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

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

6.1.1 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

6.1.2 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 6.1.2.1
Handling by CPU	The CPU handles the exception request.	Section 6.1.2.2
Branch to ISR	The CPU branches to the corresponding interrupt service routine (ISR).	
Execution of ISR	Necessary processing is executed.	Section 6.1.2.3
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 6.1.2.4

6.1.2.1 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 "6.5 Interrupts".

(2) Exception detection

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

Table 6-1 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 6-1 Exception Types and Priority

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

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

Note 2: **External interrupts have different sources and numbers in each product. For details, see "6.5.1.5 List of Interrupt Sources".**

(3) Priority setting

- Priority levels

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

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

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

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

- Priority grouping

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

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

The Table 6-2 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 6-2 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".**

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

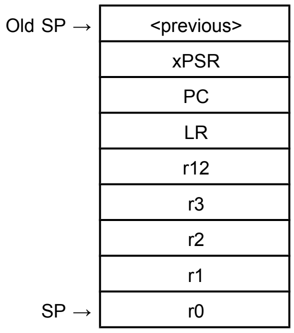
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

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

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

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



(2) Fetching an ISR

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

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

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

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

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

6.1.2.3 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 "6.5 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.

6.1.2.4 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.

6.2 Reset Exceptions

Reset exceptions are generated from the following five sources.

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

- External reset pin

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

- Reset exception by WDT

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

- Reset exception by SYSRESETREQ

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

- Reset exception by POR

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

- Reset exception by OFD

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

6.3 Non-Maskable Interrupts (NMI)

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

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

- Non-maskable interrupt by WDT

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

- Non-maskable interrupt by VLTD

  The Voltage Level Detector (VLTD) has a non-maskable interrupt generating feature. For details, see the chapter on the VLTD.

6.4 SysTick

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

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

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

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

6.5 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.

6.5.1 Interrupt Sources

6.5.1.1 Interrupt Route

Figure 6-1 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).

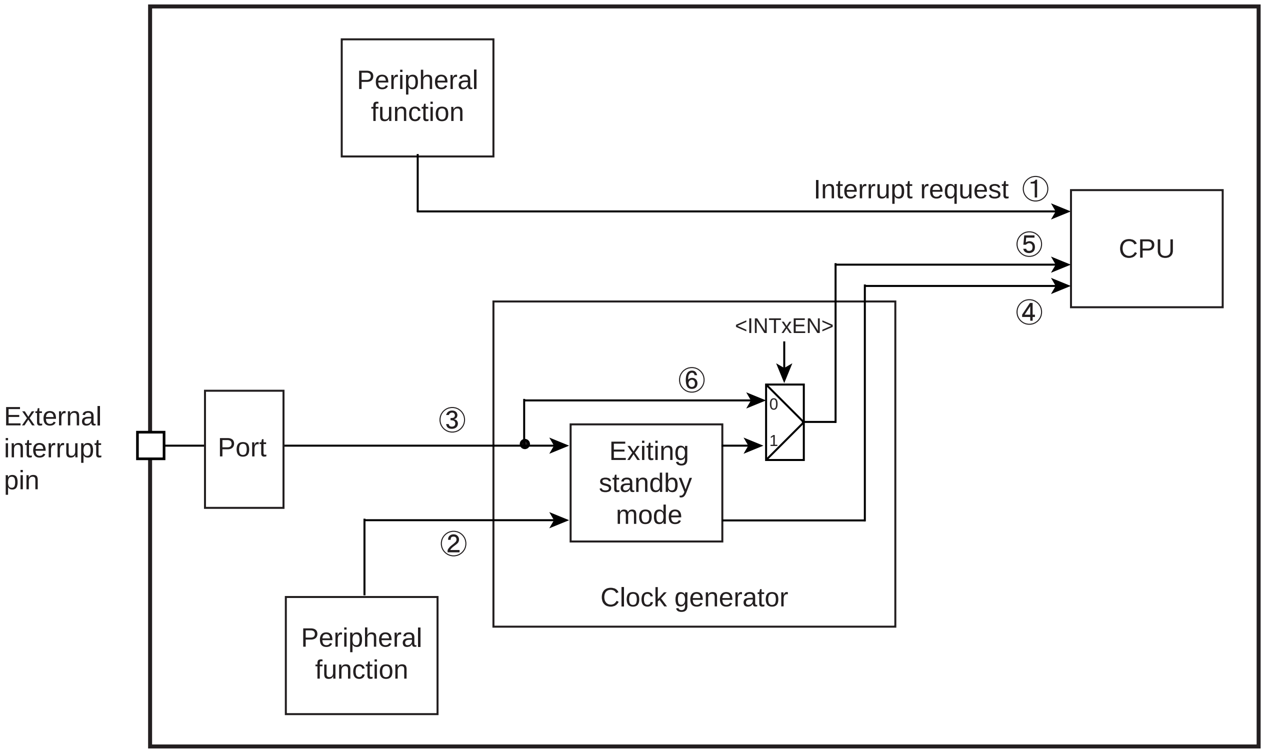


Figure 6-1 Interrupt Route

6.5.1.2 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.

6.5.1.3 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.

6.5.1.4 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 6-1 Interrupt Route"), input signals from the external interrupt pins are directly sent to the CPU. Since the CPU recognizes "High" input as an interrupt, interrupts occur if corresponding interrupts are enabled by the CPU as inputs are being disabled.

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

6.5.1.5 List of Interrupt Sources

Table 6-3 shows the list of interrupt sources.

Table 6-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt pin 0	Selectable	CGIMCGA
1	INT1	Interrupt pin 1		
2	INT2	Interrupt pin 2		
3	INT3	Interrupt pin 3		
4	INT4	Interrupt pin 4		CGIMCGB
5	INT5	Interrupt pin 5		
6	INTRX0	Serial reception (channel 0)		
7	INTTX0	Serial transmission (channel 0)		
8	INTRX1	Serial reception (channel 1)		
9	INTTX1	Serial transmission (channel 1)		
10	INTSSP0	SSP serial interface (channel 0)		
11	INTSSP1	SSP serial interface (channel 1)		
12	INTEMG0	PMD EMG Interrupt (channel 0)		
13	INTEMG1	PMD EMG Interrupt (channel 1)		
14	INTSBI0	Serial bus interface 0		
15	INTSBI1	Serial bus interface 1		
16	INTADPD0	ADC conversion triggered by PMD0 is finished		
17	INTRTC	Realtime clock interrupt	Falling edge	CGIMCGE
18	INTADPD1	ADC conversion triggered by PMD1 is finished		
19	INTRMCRX	Remote Controller reception interrupt	Rising edge	CGIMCGE
20	INTTB00	16-bit TMRB compare match detection 0 / overflow (channel 0)		
21	INTTB01	16-bit TMRB compare match detection 1(channel 0)		
22	INTTB10	16-bit TMRB compare match detection 0 / overflow (channel 1)		
23	INTTB11	16-bit TMRB compare match detection 1(channel 1)		
24	INTTB40	16-bit TMRB compare match detection 0 / overflow (channel 4)		
25	INTTB41	16-bit TMRB compare match detection 1(channel 4)		
26	INTTB50	16-bit TMRB compare match detection 0 / overflow (channel 5)		
27	INTTB51	16-bit TMRB compare match detection 1(channel 5)		
28	INTPMD0	PMD PWM Interrupt (channel 0)		
29	INTPMD1	PMD PWM Interrupt (channel 1)		
30	INTCAP00	16-bit TMRB input capture 0 (channel 0)		
31	INTCAP01	16-bit TMRB input capture 1 (channel 0)		
32	INTCAP10	16-bit TMRB input capture 0 (channel 1)		
33	INTCAP11	16-bit TMRB input capture 1 (channel 1)		
34	INTCAP40	16-bit TMRB input capture 0 (channel 4)		
35	INTCAP41	16-bit TMRB input capture 1 (channel 4)		
36	INTCAP50	16-bit TMRB input capture 0 (channel 5)		
37	INTCAP51	16-bit TMRB input capture 1 (channel 5)		
38	INT6	Interrupt pin 6	Selectable	CGIMCGB
39	INT7	Interrupt pin 7		
40	INTRX2	Serial reception (channel 2)		

Table 6-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
41	INTTX2	Serial transmission (channel 2)		
42	INTADCP0	ADC conversion monitoring function interrupt 0		
43	INTADCP1	ADC conversion monitoring function interrupt 1		
44	INTRX4	Serial reception (channel 4)		
45	INTTX4	Serial transmission (channel 4)		
46	INTTB20	16-bit TMRB compare match detection 0 / overflow (channel 2)		
47	INTTB21	16-bit TMRB compare match detection 1(channel 2)		
48	INTTB30	16-bit TMRB compare match detection 0 / overflow (channel 3)		
49	INTTB31	16-bit TMRB compare match detection 1(channel 3)		
50	INTCAP20	16-bit TMRB input capture 0 (channel 2)		
51	INTCAP21	16-bit TMRB input capture 1 (channel 2)		
52	INTCAP30	16-bit TMRB input capture 0 (channel 3)		
53	INTCAP31	16-bit TMRB input capture 1 (channel 3)		
54	INTADSFT	ADC conversion started by software is finished		
55	Reserved	Reserved		
56	INTADTMR	ADC conversion triggered by timer is finished		
57	Reserved	Reserved		
58	INT8	Interrupt pin 8	Selectable	CGIMCGC
59	INT9	Interrupt pin 9		
60	INTA	Interrupt pin A		
61	INTB	Interrupt pin B		
62	INTENC0	Encoder input interrupt (channel 0)		
63	INTENC1	Encoder input interrupt (channel 1)		
64	INTRX3	Serial reception (channel 3)		
65	INTTX3	Serial transmission (channel 3)		
66	INTTB60	16-bit TMRB compare match detection 0 / overflow (channel 6)		
67	INTTB61	16-bit TMRB compare match detection 1(channel 6)		
68	INTTB70	16-bit TMRB compare match detection 0 / overflow (channel 7)		
69	INTTB71	16-bit TMRB compare match detection 1(channel 7)		
70	INTCAP60	16-bit TMRB input capture 0 (channel 6)		
71	INTCAP61	16-bit TMRB input capture 1 (channel 6)		
72	INTCAP70	16-bit TMRB input capture 0 (channel 7)		
73	INTCAP71	16-bit TMRB input capture 1 (channel 7)		
74	INTC	Interrupt pin C	Selectable	CGIMCGD
75	INTD	Interrupt pin D		
76	INTE	Interrupt pin E		
77	INTF	Interrupt pin F		
78	INTDMACERR	DMA transfer error		
79	INTDMACTC	DMA end of transfer		
80	INTMTTB00	MPT IGBT period/ compare match detection 0/ Over flow (channel 0)		
81	INTMTTB01	MPT IGBT trigger/ compare match detection 1 (channel 0)		
82	INTMTTB10	MPT IGBT period/ compare match detection 0/ Over flow (channel 1)		

Table 6-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
83	INTMTTB11	MPT IGBT trigger/ compare match detection 1 (channel 1)		
84	INTMTTB20	MPT IGBT period/ compare match detection 0/ Over flow (channel 2)		
85	INTMTTB21	MPT IGBT trigger/ compare match detection 1 (channel 2)		
86	INTMTCAP00	MPT input capture 0 (channel 0)		
87	INTMTCAP01	MPT input capture 1 (channel 0)		
88	INTMTCAP10	MPT input capture 0 (channel 1)		
89	INTMTCAP11	MPT input capture 1 (channel 1)		
90	INTMTCAP20	MPT input capture 0 (channel 2)		
91	INTMTCAP21	MPT input capture 1 (channel 2)		
92	INTMTEMG0	MPT IGBT EMG interrupt (channel 0)		
93	INTMTEMG1	MPT IGBT EMG interrupt (channel 1)		
94	INTMTEMG2	MPT IGBT EMG interrupt (channel 2)		

6.5.1.6 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[2:0]> bits. You must set the active level for interrupt requests from each peripheral func-tion as shown in Table 6-3.

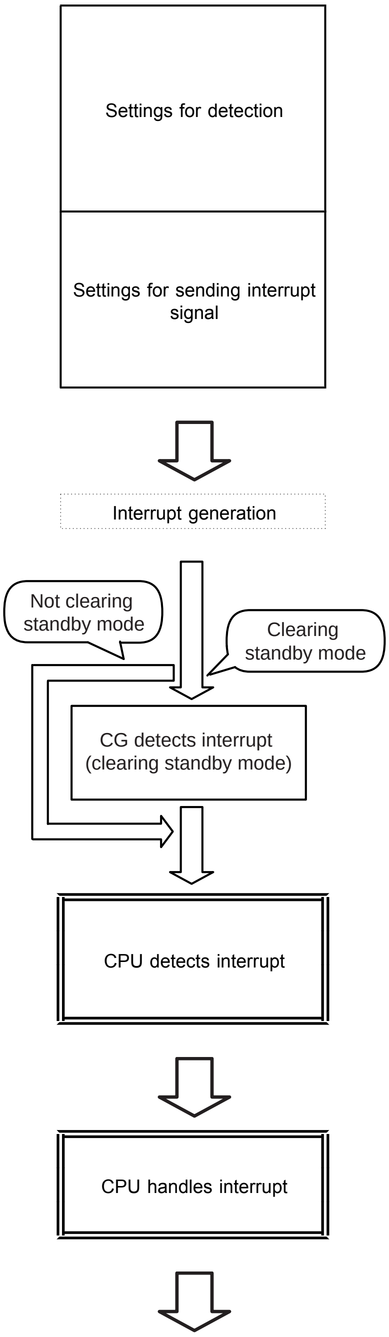
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

6.5.2 Interrupt Handling

6.5.2.1 Flowchart

The following shows how an interrupt is handled.

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



Processing	Details	See
Settings for detection	Set the relevant NVIC registers for detecting interrupts. Set the clock generator as well if each interrupt source is used to clear a standby mode. oCommon setting NVIC registers oSetting to clear standby mode Clock generator	"6.5.2.2 Preparation"
Settings for sending interrupt signal	Execute an appropriate setting to send the interrupt signal depending on the interrupt type. oSetting for interrupt from external pin Port oSetting for interrupt from peripheral function Peripheral function (See the chapter of each peripheral function for details.)	
Interrupt generation	An interrupt request is generated.	
CG detects interrupt (clearing standby mode)	Interrupt lines used for clearing a standby mode are connected to the CPU via the clock generator.	"6.5.2.3 Detection by Clock Generator"
CPU detects interrupt	The CPU detects the interrupt. If multiple interrupt requests occur simultaneously, the interrupt request with the highest priority is detected according to the priority order.	"6.5.2.4 Detection by CPU"
CPU handles interrupt	The CPU handles the interrupt. The CPU pushes register contents to the stack before entering the ISR.	"6.5.2.5 CPU processing"

Processing	Details	See
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"6.5.2.6 Interrupt Service Routine (ISR)"
Return to preceding program	Configure to return to the preceding program of the ISR.	

6.5.2.2 Preparation

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

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

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

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

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

(1) Disabling interrupt by CPU

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

Use "MSR" instruction to set this register.

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

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

(2) CPU registers setting

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

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

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

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

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

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

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

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

Note: **x: port number / m: corresponding bit / n: function register number**
In modes other than STOP mode, setting PxIE to enable input enables the corresponding interrupt input regardless of the PxFR setting. Be careful not to enable interrupts that are not used. Also, be aware of the description of "6.5.1.4 Precautions when using external interrupt pins".

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

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

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

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

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

Note: **m: corresponding bit**

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "6.6.3.6 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"6.5.1.4 Precautions when using external interrupt pins".

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

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

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

Note 1: **m : corresponding bit**

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

6.5.2.3 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.

6.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

6.5.2.5 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.

6.5.2.6 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.

6.6 Exception/Interrupt-Related Registers

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

6.6.1 Register List

NVIC registers Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Interrupt Set-Enable Register 2	0x0104
Interrupt Set-Enable Register 3	0x0108
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Clear-Enable Register 3	0x0188
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Set-Pending Register 3	0x0208
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Clear-Pending Register 3	0x0288
Interrupt Priority Register	0x0400 to 0x045C
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18, 0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

Clock generator registers Base Address = 0x4004_0200

Register name		Address
CG Interrupt Request Clear Register	CGICRCG	0x0014
Reset Flag Register	CGRSTFLG	0x0018
NMI Flag Register	CGNMIFLG	0x001C
CG Interrupt Mode Control Register A	CGIMCGA	0x0020
CG Interrupt Mode Control Register B	CGIMCGB	0x0024
CG Interrupt Mode Control Register C	CGIMCGC	0x0028
CG Interrupt Mode Control Register D	CGIMCGD	0x002C
CG Interrupt Mode Control Register E	CGIMCGE	0x0030

6.6.2 NVIC Registers

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

6.6.2.2 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".

6.6.2.3 SysTick Current Value Register

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

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

6.6.2.4 SysTick Calibration Value Register

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

Bit	Bit Symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.
29-24	−	R	Read as 0.
23-0	TENMS	R	Calibration value Reload value to use for 10 ms timing (0x9C4). (Note)

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

6.6.2.5 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 "6.5.1.5 List of Interrupt Sources".**

6.6.2.6 Interrupt Set-Enable Register 2

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.7 Interrupt Set-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	SETENA (Interrupt 94)	SETENA (Interrupt 93)	SETENA (Interrupt 92)	SETENA (Interrupt 91)	SETENA (Interrupt 90)	SETENA (Interrupt 89)	SETENA (Interrupt 88)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA (Interrupt 87)	SETENA (Interrupt 86)	SETENA (Interrupt 85)	SETENA (Interrupt 84)	SETENA (Interrupt 83)	SETENA (Interrupt 82)	SETENA (Interrupt 81)	SETENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 79)	SETENA (Interrupt 78)	SETENA (Interrupt 77)	SETENA (Interrupt 76)	SETENA (Interrupt 75)	SETENA (Interrupt 74)	SETENA (Interrupt 73)	SETENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 71)	SETENA (Interrupt 70)	SETENA (Interrupt 69)	SETENA (Interrupt 68)	SETENA (Interrupt 67)	SETENA (Interrupt 66)	SETENA (Interrupt 65)	SETENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-0	SETENA	R/W	Interrupt number [94:64] [Write] 1: Enable [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.8 Interrupt Clear-Enable Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.9 Interrupt Clear-Enable Register 2

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.10 Interrupt Clear-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	CLRENA (Interrupt 94)	CLRENA (Interrupt 93	CLRENA (Interrupt 92)	CLRENA (Interrupt 91)	CLRENA (Interrupt 90)	CLRENA (Interrupt89)	CLRENA (Interrupt 88)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA (Interrupt 87)	CLRENA (Interrupt 86)	CLRENA (Interrupt 85)	CLRENA (Interrupt 84)	CLRENA (Interrupt 83)	CLRENA (Interrupt 82)	CLRENA (Interrupt 81)	CLRENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 79)	CLRENA (Interrupt 78)	CLRENA (Interrupt 77)	CLRENA (Interrupt 76)	CLRENA (Interrupt 75)	CLRENA (Interrupt 74)	CLRENA (Interrupt 73)	CLRENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 71)	CLRENA (Interrupt 70)	CLRENA (Interrupt 69)	CLRENA (Interrupt 68)	CLRENA (Interrupt 67)	CLRENA (Interrupt 66)	CLRENA (Interrupt 65)	CLRENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-0	CLRENA	R/W	Interrupt number [94:64] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.11 Interrupt Set-Pending Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.12 Interrupt Set-Pending Register 2

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

Bit	Bit Symbol	Type	Function
31-0	SETPEND	R/W	Interrupt number [63:32] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.13 Interrupt Set-Pending Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	SETPEND (Interrupt 94)	SETPEND (Interrupt 93)	SETPEND (Interrupt 92)	SETPEND (Interrupt 91)	SETPEND (Interrupt 90)	SETPEND (Interrupt 89)	SETPEND (Interrupt 88)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (Interrupt 87)	SETPEND (Interrupt 86)	SETPEND (Interrupt 85)	SETPEND (Interrupt 84)	SETPEND (Interrupt 83)	SETPEND (Interrupt 82)	SETPEND (Interrupt 81)	SETPEND (Interrupt 80)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 79)	SETPEND (Interrupt 78)	SETPEND (Interrupt 77)	SETPEND (Interrupt 76)	SETPEND (Interrupt 75)	SETPEND (Interrupt 74)	SETPEND (Interrupt 73)	SETPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 71)	SETPEND (Interrupt 70)	SETPEND (Interrupt 69)	SETPEND (Interrupt 68)	SETPEND (Interrupt 67)	SETPEND (Interrupt 66)	SETPEND (Interrupt 65)	SETPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-0	SETPEND	R/W	Interrupt number [94:64] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Clear and Interrupt Set-Pending Register bit by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.14 Interrupt Clear-Pending Register 1

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

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

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.15 Interrupt Clear-Pending Register 2

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

Bit	Bit Symbol	Type	Function
31-0	CLRPEND	R/W	Interrupt number [63:32] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.16 Interrupt Clear-Pending Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	CLRPEND (Interrupt 94)	CLRPEND (Interrupt 93)	CLRPEND (Interrupt 92)	CLRPEND (Interrupt 91)	CLRPEND (Interrupt 90)	CLRPEND (Interrupt 89)	CLRPEND (Interrupt 88)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND (Interrupt 87)	CLRPEND (Interrupt 86)	CLRPEND (Interrupt 85	CLRPEND (Interrupt 84)	CLRPEND (Interrupt 83)	CLRPEND (Interrupt 82)	CLRPEND (Interrupt 81)	CLRPEND (Interrupt 80)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 79)	CLRPEND (Interrupt 78)	CLRPEND (Interrupt 77)	CLRPEND (Interrupt 76)	CLRPEND (Interrupt 75)	CLRPEND (Interrupt 74)	CLRPEND (Interrupt 73)	CLRPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 71)	CLRPEND (Interrupt 70)	CLRPEND (Interrupt 69)	CLRPEND (Interrupt 68)	CLRPEND (Interrupt 67)	CLRPEND (Interrupt 66)	CLRPEND (Interrupt 65)	CLRPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-0	CLRPEND	R/W	Interrupt number [94:64] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

Note: **For descriptions of interrupts and interrupt numbers, see Section "6.5.1.5 List of Interrupt Sources".**

6.6.2.17 Interrupt Priority Register

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

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

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E434	PRI_55	PRI_54	PRI_53	PRI_52
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E44C	PRI_79	PRI_78	PRI_77	PRI_76
0xE000_E450	PRI_83	PRI_82	PRI_81	PRI_80
0xE000_E454	PRI_87	PRI_86	PRI_85	PRI_84
0xE000_E458	PRI_91	PRI_90	PRI_89	PRI_88
0xE000_E45C	-	PRI_94	PRI_93	PRI_92

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.

6.6.2.18 Vector Table Offset Register

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

Bit	Bit Symbol	Type	Function
31-30	−	R	Read as 0.
29	TBLBASE	R/W	Table base The vector table is in: 0: Code space 1: SRAM space
28-7	TBLOFF	R/W	Offset value Set the offset value from the top of the space specified in TBLBASE. The offset must be aligned based on the number of exceptions in the table.This means that the minimum alignment is 32 words that you can use for up to 16 interrupts.For more interrupts, you must adjust the alignment by rounding up to the next power of two.
6-0	−	R	Read as 0.

6.6.2.19 Application Interrupt and Reset Control Register

	31	30	29	28	27	26	25	24
bit symbol	VECTKEY/VECTKEYSTAT							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	VECTKEY/VECTKEYSTAT							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENDIANESS	-	-	-	-	PRIGROUP		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SYSRESET REQ	VECTCLR ACTIVE	VECTRESET
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	VECTKEY (Write)/ VECTKEY-STAT(Read)	R/W	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05.
15	ENDIANESS	R/W	Endianness bit:(Note1) 1: big endian 0: little endian
14-11	−	R	Read as 0.
10-8	PRIGROUP	R/W	Interrupt priority grouping 000: seven bits of pre-emption priority, one bit of subpriority 001: six bits of pre-emption priority, two bits of subpriority 010: five bits of pre-emption priority, three bits of subpriority 011: four bits of pre-emption priority, four bits of subpriority 100: three bits of pre-emption priority, five bits of subpriority 101: two bits of pre-emption priority, six bits of subpriority 110: one bit of pre-emption priority, seven bits of subpriority 111: no pre-emption priority, eight bits of subpriority The bit configuration to split the interrupt priority register <PRI_n> into pre-emption priority and sub priority.
7-3	−	R	Read as 0.
2	SYSRESET REQ	R/W	System Reset Request. 1=CPU outputs a SYSRESETREQ signal. (note2)
1	VECTCLR ACTIVE	R/W	Clear active vector bit 1: clear all state information for active NMI, fault, and interrupts 0: do not clear. This bit self-clears. It is the responsibility of the application to reinitialize the stack.
0	VECTRESET	R/W	System Reset bit 1: reset system 0: do not reset system Resets the system, with the exception of debug components (FPB, DWT and ITM) by setting "1" and this bit is also zero cleared.

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

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

6.6.2.20 System Handler Priority Register

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

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

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

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

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

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

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

6.6.2.21 System Handler Control and State Register

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

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

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

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

6.6.3 Clock generator registers

6.6.3.1 CGIMCGA(CG Interrupt Mode Control Register A)

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

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

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

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

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

6.6.3.2 CGIMCGB(CG Interrupt Mode Control Register B)

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

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

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

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

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

6.6.3.3 CGIMCGC(CG Interrupt Mode Control Register C)

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

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

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

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

6.6.3.4 CGIMCGD(CG Interrupt Mode Control Register D)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	EMCGF[2:0]	R/W	active level setting of INTF standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
27-26	EMSTF[1:0]	R	active level of INTF standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
25	−	R	Reads as undefined.
24	INTFEN	R/W	INTF clear input 0:Disable 1: Enable
23	−	R	Read as 0.
22-20	EMCGE[2:0]	R/W	active level setting of INTE standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges
19-18	EMSTE[1:0]	R	active level of INTE standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
17	−	R	Reads as undefined.
16	INTEEN	R/W	INTE clear input 0:Disable 1: Enable
15	−	R	Read as 0.
14-12	EMCGD[2:0]	R/W	active level setting of INTD standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges

Bit	Bit Symbol	Type	Function
11-10	EMSTD[1:0]	R	active level of INTD standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INTDEN	R/W	INTD clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCGC[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 edges
3-2	EMSTC[1:0]	R	active level of INTC standby clear request 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INTCEN	R/W	INTC clear input 0:Disable 1: Enable

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

6.6.3.5 CGIMCGE(CG Interrupt Mode Control Register E)

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

Bit	Bit Symbol	Type	Function
31	−	R	Read as 0.
30-28	−	R/W	Write any value.
27-25	−	R	Read as 0.
24	−	R/W	Write as 0.
23	−	R	Read as 0.
22-20	−	R/W	Write any value.
19-17	−	R	Read as 0.
16	−	R/W	Write as 0.
15	−	R	Read as 0.
14-12	EMCGRMCRX [2:0]	R/W	active level setting of INTRMCRX standby clear request. (101 to 111: setting prohibited) 011: Rising edge
11-10	EMSTRMCRX [1:0]	R	active level of INTRMCRX standby clear request 00: − 01: Rising edge 10: − 11: −
9	−	R	Reads as undefined.
8	INTRMCRXEN	R/W	INTRMCRX clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCGRTC [2:0]	R/W	active level setting of INTRTC standby clear request. (101 to 111: setting prohibited) 010: Falling edge
3-2	EMSTRTC[1:0]	R	active level of INTRTC standby clear request 00: − 01: − 10: Falling edge 11: −
1	−	R	Reads as undefined.
0	INTRTCEN	R/W	INTRTC clear input 0:Disable 1: Enable

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

6.6.3.6 CGICRCG(CG Interrupt Request Clear Register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000: INT0 0_1000: INT8 1_0000: INTRTC 0_0001: INT1 0_1001: INT9 1_0001: INTRMCRX 0_0010: INT2 0_1010: INTA 0_0011: INT3 0_1011: INTB 0_0100: INT4 0_1100: INTC 0_0101: INT5 0_1101: INTD 0_0110: INT6 0_1110: INTE 0_0111: INT7 0_1111: INTF 1_00010 to 1_1111: setting prohibited. Read as 0.

6.6.3.7 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	-	-	-	-	-	NMIFLG2	-	NMIFLG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	NMIFLG2	R	NMI source generation flag 0: not applicable 1: generated from voltage detection
1	−	R	Reads as undefined.
0	NMIFLG0	R	NMI source generation flag 0: not applicable 1: generated from WDT

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

6.6.3.8 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	-	WDTRSTF	PINRSTF	PONRSTF
After power-on reset	0	0	0	0	0	0	0	1

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

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

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

7. DMA Controller (DMAC)

7.1 Function Overview

The table below lists its major functions.

Table 7-1 DMA controller functions

Item	Function		Overview
Number of channels	2ch (1 Unit)		
	Hardware start		Supports DMA requests for peripheral IPs. (Refer to Table 7-3)
	Software start		Started with a write operation to the DMACx-SoftBReq register.
Bus master	32bit × 1 (AHB)		
Priority	(High) DMA ch0 to DMA ch1 (Low)		Fixed by hardware
FIFO	4word × 2ch		
Bus width	8/16/32bit		Settable individually for transfer source and destination.
Burst size	1/4/8/16/32/64/128/256		
Number of transfers	up to 4095		
Address	Transfer source address	incr / no-incr	It is possible to specify whether Source and Destination addresses should increment or should not increment (should be fixed). (Address wrapping is not supported.)
	Transfer destination address	incr / no-incr	
Endian	Only little endian is supported.		
Transfer type	Memory → peripheral circuit (register) Peripheral circuit (register) → memory Memory → memory (note 2)		When "memory → memory" is selected, hardware startup by DMA is not supported. See the DMACCxConfiguration register for more information.
Interrupt function	Transfer end interrupt Error interrupt		
Special Function	Scatter/gather function		

Note 1: 1 word = 32 bits

Note 2: Following transfer type is not supported : From Peripheral circuit (register) to Peripheral circuit (register)

7.2 DMA transfer type

Table 7-2 DMA transfer type

	DMA direction	DMA request circuit	Support DMA request (Note2)	Other condition
1	Memory → peripheral circuit	peripheral circuit (Destination)	Burst request	In case of 1word transmission, set to the "1" for burst size of DMA controller.
2	Peripheral circuit → memory	peripheral circuit (Source)	Burst request / single request (Note1)	If the amount of data transfer is not an integer multiples of the burst size, both burst and single transfers can be used. If the amount of data transfer defined with DMA controller is the same size of the burst size or more, the single request is ignored and the burst transfer is taken place. If the amount of data transfer defined with DMA controller is less than the burst size, the single burst transfer is taken place.
3	Memory → memory	DMAC	−	No DMA request occurs. When DMA circuit is enabled, data transfer starts. (Memory to memory transfer is selected and DMACCxConfiguration<E> = "1" is set.) When all data transfer is completed or DMA channel is disabled, data transfer stops.

Note 1: SSP: Peripheral circuit corresponding to the single request

Note 2: For supported DMA requests, refer to later pages.

7.3 Block diagram

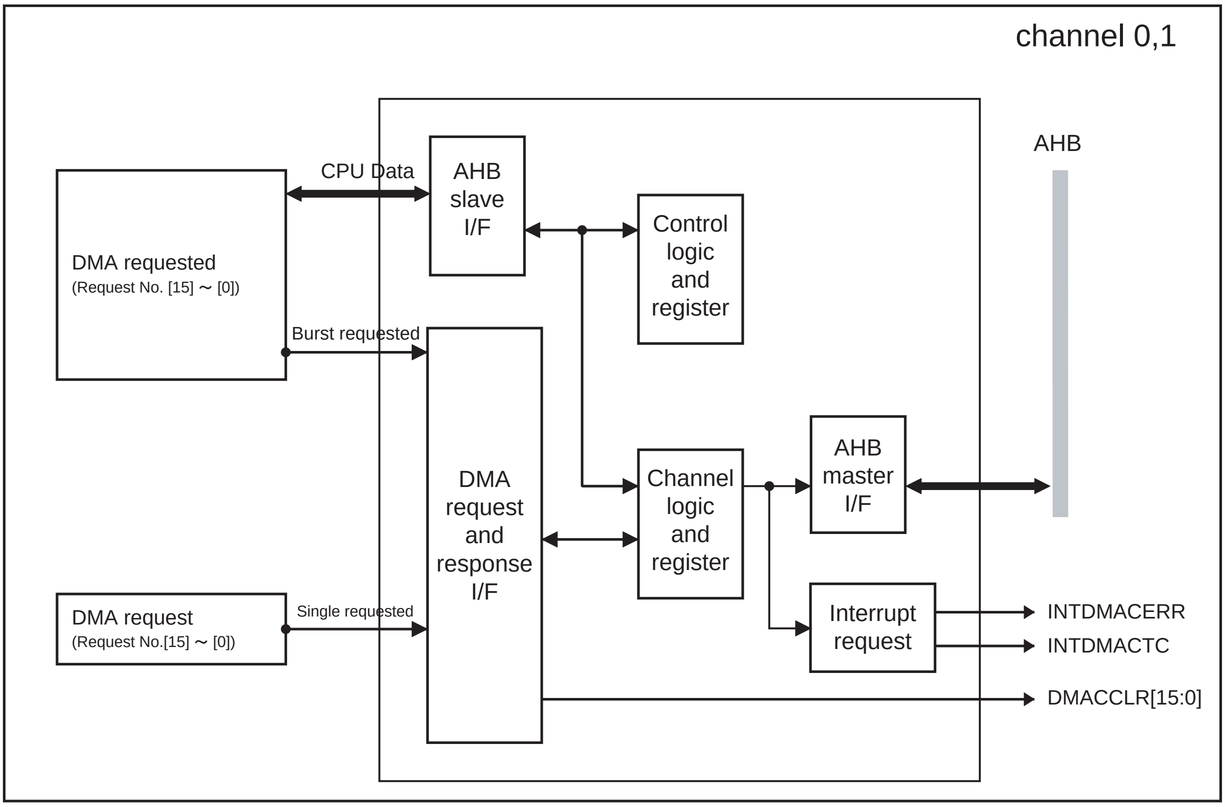


Figure 7-1 DMAC Block Diagram

Table 7-3 DMA request number chart

DMA request No.	Corresponding peripheral	
	Burst	Single
0	SIO0 Reception/Transmission	–
1	SIO1 Reception / Transmission	–
2	SIO2 Reception / Transmission	–
3	SIO3 Reception / Transmission	–
4	SIO4 Reception / Transmission	–
5	SSP0 Transmission	–
6	SSP0 Reception	SSP0 Reception
7	SSP1 Transmission	–
8	SSP1 Reception	SSP1 Reception
9	–	–
10	–	–
11	–	–
12	–	–
13	–	–
14	–	–
15	–	–

7.4 Description of Registers

7.4.1 DMAC register list

The following lists the each unit and address:

Base Address = 0x4008_0000

Register Name		Address (Base+)
DMAC Interrupt Status Register	DMACIntStaus	0x0000
DMAC Interrupt Terminal Count Status Register	DMACIntTCStatus	0x0004
DMAC Interrupt Terminal Count Clear Register	DMACIntTCClear	0x0008
DMAC Interrupt Error Status Register	DMACIntErrorStatus	0x000C
DMAC Interrupt Error Clear Register	DMACIntErrClr	0x0010
DMAC Raw Interrupt Terminal Count Status Register	DMACRawIntTCStatus	0x0014
DMAC Raw Error Interrupt Status Register	DMACRawIntErrorStatus	0x0018
DMAC Enabled Channel Register	DMACEnbldChns	0x001C
DMAC Software Burst Request Register	DMACSoftBReq	0x0020
DMAC Software Single Request Register	DMACSoftSReq	0x0024
Reserved	-	0x0028
Reserved	-	0x002C
DMAC Configuration Register	DMACConfiguration	0x0030
Reserved	-	0x0034
DMAC Channel0 Source Address Register	DMACC0SrcAddr	0x0100
DMAC Channel0 Destination Address Register	DMACC0DestAddr	0x0104
DMAC Channel0 Linked List Item Register	DMACC0LLI	0x0108
DMAC Channel0 Control Register	DMACC0Control	0x010C
DMAC Channel0 Configuration Register	DMACC0Configuration	0x0110
DMAC Channel1 Source Address Register	DMACC1SrcAddr	0x0120
DMAC Channel1 Destination Address Register	DMACC1DestAddr	0x0124
DMAC Channel1 Linked List Item Register	DMACC1LLI	0x0128
DMAC Channel1 Control Register	DMACC1Control	0x012C
DMAC Channel 1 Configuration Register	DMACC1Configuration	0x0130

Note 1: Only word (32bit) access can be used in the above registers.

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

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

7.4.2 DMACIntStatus (DMAC Interrupt Status Register)

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

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

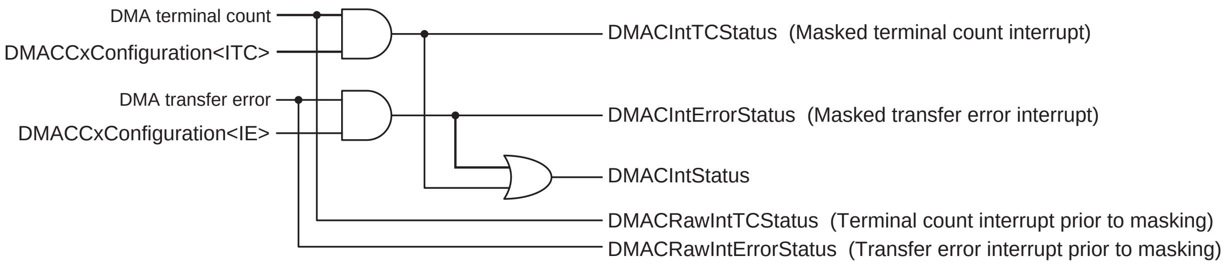

DMA terminal count
DMACCxConfiguration<ITC>
DMA transfer error
DMACCxConfiguration<IE>
DMACIntTCStatus (Masked terminal count interrupt)
DMACIntErrorStatus (Masked transfer error interrupt)
DMACIntStatus
DMACRawIntTCStatus (Terminal count interrupt prior to masking)
DMACRawIntErrorStatus (Transfer error interrupt prior to masking)

Figure 7-2 Interrupt-related block diagram

7.4.3 DMACIntTCStatus (DMAC Interrupt Terminal Count Status Register)

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

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

7.4.4 DMACIntTCClear (DMAC Interrupt Terminal Count Clear Register)

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

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntTCClear1	W	Clear DMAC channel 1 transfer end interrupt. 0 : Invalid 1 : Clear The DMACIntTCStatus<IntTCStatus1> will be cleared when "1" is written.
0	IntTCClear0	W	Clear DMAC channel 0 transfer end interrupt. 0 : Invalid 1 : Clear The DMACIntTCStatus<IntTCStatus0> will be cleared when "1" is written.

7.4.5 DMACIntErrorStatus (DMAC Interrupt Error Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	IntErrStatus1	IntErrStatus0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

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

7.4.6 DMACIntErrClr (DMAC Interrupt Error Clear Register)

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

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	IntErrClr1	W	Clear DMAC channel 1 error interrupt. 0 : Invalid 1 : Clear The DMACIntErrorStatus<IntErrStatus1> will be cleared when "1" is written.
0	IntErrClr0	W	Clear DMAC channel 0 error interrupt. 0 : Invalid 1 : Clear The DMACIntErrorStatus<IntErrStatus0> will be cleared when "1" is written.

7.4.7 DMACRawIntTCStatus (DMAC Raw Interrupt Terminal Count Status Register)

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

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

7.4.8 DMACRawIntErrorStatus (DMAC Raw Error Interrupt Status Register)

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

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	RawIntErrS1	R	Status of DMAC channel 1 before error interrupt generation is enabled. 0 : Interrupt not requested 1 : Interrupt requested
0	RawIntErrS0	R	Status of DMAC channel 0 before error interrupt generation is enabled. 0 : Interrupt not requested 1 : Interrupt requested

7.4.9 DMACEnbldChns (DMAC Enabled Channel Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	EnabledCH1	EnabledCH0
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Description
31-2	-	W	Write as zero.
1	EnabledCH1	R	DMAC channel 1 enable status. 0 : The bits of the appropriate channel are cleared when DMA transfer is complete. 1 : Channel 1 is enabled.
0	EnabledCH0	R	DMAC channel 0 enable status. 0 : The bits of the appropriate channel are cleared when DMA transfer is complete. 1 :Channel 0 is enabled.

7.4.10 DMACSoftBReq (DMAC Software Burst Request Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	SoftBReq8
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	SoftBReq7	SoftBReq6	SoftBReq5	SoftBReq4	SoftBReq3	SoftBReq2	SoftBReq1	SoftBReq0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-9	-	W	Write as zero.
8	SoftBReq8	R/W	DMA burst request by software (Request No. [8]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
7	SoftBReq7	R/W	DMA burst request by software (Request No. [7]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
6	SoftBReq6	R/W	DMA burst request by software (Request No. [6]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
5	SoftBReq5	R/W	DMA burst request by software (Request No. [5]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
4	SoftBReq4	R/W	DMA burst request by software (Request No. [4]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
3	SoftBReq3	R/W	DMA burst request by software (Request No. [3]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
2	SoftBReq2	R/W	DMA burst request by software (Request No. [2]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
1	SoftBReq1	R/W	DMA burst request by software (Request No. [1])

Bit	Bit Symbol	Type	Description
			When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs
0	SoftBReq0	R/W	DMA burst request by software (Request No. [0]) When read 0 : DMA burst data transfer is stopping 1 : DMA burst data transfer is operating When write 0 : Invalid 1 : DMA burst request occurs

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

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

Note 2: Refer to "Table 7-3 DMA request number chart" for DMA request number.

7.4.11 DMACSoftSReq (DMAC Software Single Request Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	SoftSReq8
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	SoftSReq6	-	-	-	-	-	-
After reset	Undefined	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Description
31-9	-	W	Write as zero.
8	SoftSReq8	R/W	DMA single request by software (Request No. [8]). When read 0 : DMA single burst data transfer is stopping 1 : DMA single burst data transfer is operating When write 0 : Invalid 1 : DMA single burst request occurs
7	-	W	Write as zero.
6	SoftSReq6	R/W	DMA single request by software (Request No. [6]). When read 0 : DMA single burst data transfer is stopping 1 : DMA single burst data transfer is operating When write 0 : Invalid 1 : DMA single burst request occurs
5-0	-	W	Write as zero.

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

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

Note 2: Refer to "Table 7-3 DMA request number chart" for DMA request number.

7.4.12 DMACConfiguration (DMAC Configuration Register)

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

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

7.4.13 DMACCxSrcAddr (DMAC Channelx Source Address Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Description</th></tr>
<tr><td>31-0</td><td>SrcAddr[31:0]</td><td>R/W</td><td>Sets a DMA transfer source address.
Confirm the transfer destination of memory, the bit width of IP and addresses before setting the DMA transfer.
Depending on the bit width of transfer sources, the following restrictions are given.
<table>
<tr><th>Bit width of transfer source
DMACCxControl<Swidth[2:0]&gt;</th><th>Setting of LSB address</th></tr>
<tr><td>000 :BYTE (8bits)</td><td>No restrictions</td></tr>
<tr><td>001 :Half-word (16bits)</td><td>Set the number by a factor of two
(0x00,0x02,0x04,0x06,0x08,0x0A,0x0C…)</td></tr>
<tr><td>010 :WORD (32bits)</td><td>Set the number by a factor of four (0x00,0x04,0x08,0x0C…)</td></tr>
</table></td></tr>
</table>

Because enabling channel"x" (DMACxCConfiguration<E> = "1") updates the data written in the registers, set DMACxCSrcAddr before enabling the channels.

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

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

7.4.14 DMACCxDestAddr (DMAC Channelx Destination Address Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Description</th></tr>
<tr><td>31-0</td><td>DestAddr[31:0]</td><td>R/W</td><td>Sets a DMA transfer source address.
Confirm the transfer destination of memory, the bit width of IP and addresses before setting the DMA transfer.
Depending on the bit width of transfer destinations, the following restrictions are given.
<table>
<tr><th>Bit width of transfer destination
DMACCxControl<Dwidth[2:0]></th><th>Setting of LSB address</th></tr>
<tr><td>000 :BYTE (8bits)</td><td>No restrictions</td></tr>
<tr><td>001 :Half-word (16bits)</td><td>Set the number by a factor of two (0x00,0x02,0x04,0x06,0x08,0x0A,0x0C…)</td></tr>
<tr><td>010 :WORD (32bits)</td><td>Set the number by a factor of four (0x00,0x04,0x08,0x0C…)</td></tr>
</table></td></tr>
</table>

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

7.4.15 DMACCxLLI (DMAC Channelx Linked List Item Register)

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

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

For <LLI> detailed operation, see "7.5 Special Functions".

7.4.16 DMACCxControl (DMAC Channelx Control Register)

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

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

Bit	Bit Symbol	Type	Description
11-0	TransferSize [11:0]	R/W	Set the total number of transfers. Set the total number of data transfers in the units of data bit width (4bytes/2bytes/1byte) defined as that of the transfer source. The burst size indicates only the total amount of data to be transferred per internal DMAC request. Thus as long as the bit width of transfer source and the total number of transfers are not changed, the total amount of transfer data is not changed even if the burst size is changed. The <TransferSize> value decrements with respect to each DMA transfer until it reaches 0. On read, the number of transfers yet to be performed is read. The total number of transfers is used as the unit for the transfer source bit width. For example: When <Swidth> = "000"(8bit), the number of transfers is expressed in the units of byte. When <Swidth> = "001"(16bit), the number of transfers is expressed in the units of half word. When <Swidth> = "010"(32bit), the number of transfers is expressed in the units of word.

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

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

7.4.17 DMACCxConfiguration (DMAC Channelx Configuration Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	Halt	Active	Lock
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ITC	IE	FlowCntrl			-	DestPeripheral	
After reset	0	0	0	0	0	Undefined	0	0
	7	6	5	4	3	2	1	0
bit symbol	DestPeripheral		-	SrcPeripheral				E
After reset	0	0	Undefined	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-19	-	W	Write as zero.
18	Halt	R/W	Controls accepting a DMA request 0 : Accept a DMA request 1 : Ignore a DMA request
17	Active	R	Indicates whether data is present in the channel FIFO. 0 : No data exists in the FIFO 1 : Data exists in the FIFO
16	Lock	R/W	Sets a locked transfer (Non-divided transfer). 0 : Disable locked transfer 1: Enable locked transfer When locked transfer is enabled, as many burst transfers as specified are consecutively executed without releasing the bus. For detailed operation, see "7.5 Special Functions".
15	ITC	R/W	Terminal count interrupt enable register 0 : Disable interrupts 1 : Enable interrupts If <ITC> = "1" and DMACCxControl Register<I> = "1" are set, transfer end interrupt occurs.
14	IE	R/W	Error interrupt enable register 0 : Disable interrupts 1 : Enable interrupts
13-11	FlowCntrl[2:0]	R/W	This bit sets the transfer mode.(Note 1) <FlowCntrl[2:0]> set value — Transfer Mode 000: — Memory to Memory 001: — Memory to Peripheral 010: — Peripheral to Memory 011 to 111: — Reserved
10	-	W	Write as zero.
9-6	DestPeripheral [3:0]	R/W	Transfer destination peripheral (Note 2) 0000 to 1111 This is a DMA request peripheral number in binary. This setting will be ignored if memory is specified as the transfer destination.
5	-	W	Write as zero.

Bit	Bit Symbol	Type	Description
4-1	SrcPeripheral [3:0]	R/W	Transfer source peripheral (Note 2) 0000 to 1111 This is a DMA request peripheral number in binary. This setting will be ignored if memory is specified as the transfer source.
0	E	R/W	Channel enable 0 : Disable 1 : Enable This bit is used to enable or disable the channel. (When memory to memory transfer is set, this bit operates as a transfer start bit.) When total number of transfers of DMACCxControl register is complete (the value becomes 0), the corresponding channel is automatically cleared. If the channel is disabled during a transfer, the data in the channel of FIFO will be lost. To re-start the transfer, all channels must be initialized to reset. To stop DMA transfer temporarily, use the <Halt> bit to disable DMA requests. Poll the <Active> bit until it becomes 0, and then clear the <E> bit to disable the channel.

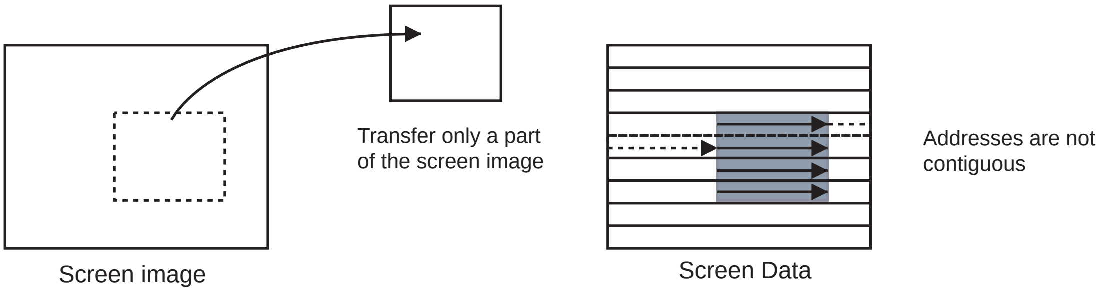
Note 1: When you selected Memory-to-Memory, hardware start triggered by DMA is not supported. Transfer is started by writing <E>= 1.

Note 2: Refer to Table 7-3 for the peripheral number of DMA request.

7.5 Special Functions

7.5.1 Scatter/gather function

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



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

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

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

1. DMACCxSrcAddr
2. DMACCxDestAddr
3. DMACCxLLI
4. DMACCxControl

It is also possible to generate interrupts in conjunction with the scatter/gather function.

If DMACCxControl<I>=1 and DMACCxConfiguration<ITC>=1 are set, DMA transfer end interrupt occurs.

An interrupt can be generated after each LLI operation by setting the terminal count interrupt enable bit of the DMACCxControl register.

In scatter/gather function, if DMA transfer end interrupt is set to occur only in the last transfer, set DMACCxControl<I> = "0" and DMACCxConfiguration<ITC> = "1" to start transfers. Then in the last DMA transfer setting flow, if <I> = "1" is set, the transfer end interrupt occurs only in the last transfer. If this bit is cleared, branch procedure added conditions can be set even if LLI is used for transfers. To clear the interrupt, the corresponding bit of DMACIntTCClear register should be controlled.

7.5.2 Linked list operation

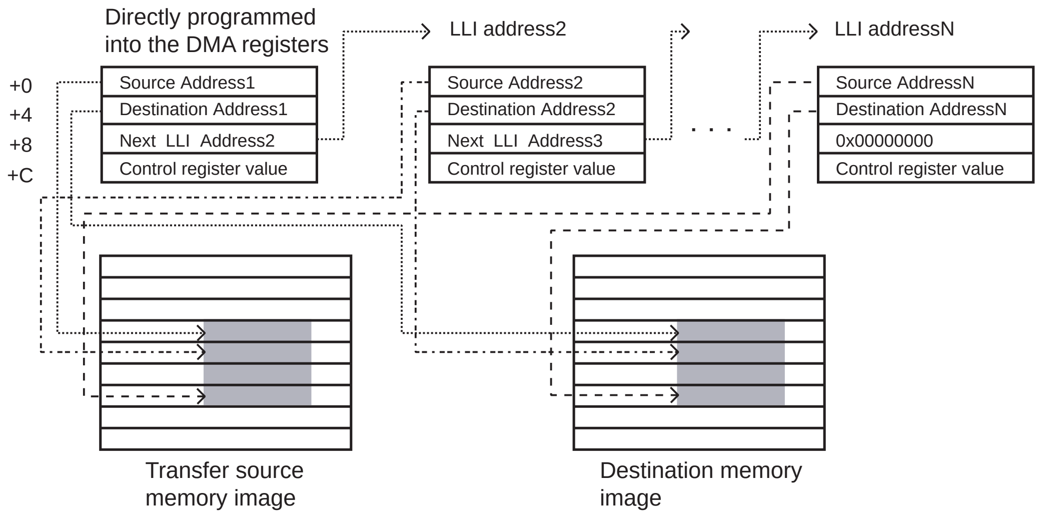
To operate the scatter/gather function, a transfer source and source data areas need to be defined by creating a set of Linked Lists first.

Each setting is called LLI (LinkedList).

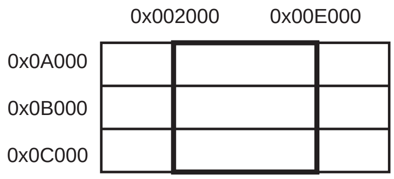
Each LLI controls the transfer of one block of data. Each LLI indicates normal DMA setting and controls transfer of successive data. Each time DMA transfer is complete, the next LLI setting will be loaded to continue the DMA operation (Daisy Chain).

An example of the setting is shown below.

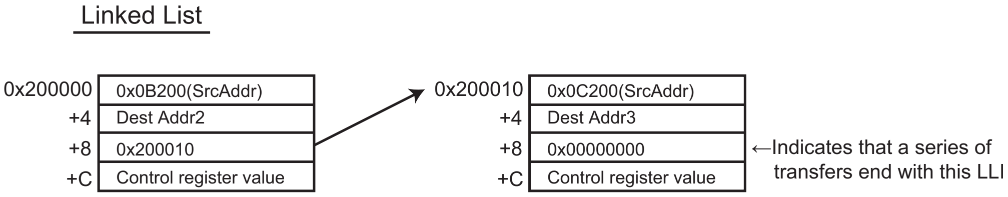
1. The first DMA transfer setting should be made directly in the DMA register.
2. The second and subsequent DMA transfer settings should be written in the addresses of the memory set in "next LLI AddressX."
3. To stop up to N'th DMA transfer, set "next LLI AddressX" to 0x0000_0000.



When transferring data in the area enclosed by the square.



	Setting register	Setting parameter
+0	DMACCxSrcAddr	:0x0A200
+4	DMACCxDestAddr	:Destination address 1
+8	DMACCxLL	:0x200000
+C	DMACCxControl	:Set the number of burst transfers and the number of transfers, etc.



8. Input / Output Ports

8.1 Port Functions

8.1.1 Function list

TMPM3U6FY/FW has 84 ports. Besides the ports function, these ports can be used as I/O pins for periph-eral functions.

Table 8-1 shows the port function table.

Table 8-1 Port Function List

Port	PIn name	Input / Output	Program-mable Pull-up Pull-down	Schmitt Input	Noise Filter	Program-mable Open-drain	Function pin
Port A							
	PA0	I/O	Pull-up Pull-down	o	o	o	TB0IN, INT3
	PA1	I/O	Pull-up Pull-down	o	−	o	TB0OUT, SCOUT
	PA2	I/O	Pull-up Pull-down	o	o	o	TB1IN, INT4
	PA3	I/O	Pull-up Pull-down	o	−	o	TB1OUT, RXIN
	PA4	I/O	Pull-up Pull-down	o	−	o	SCLK1, $\overline{\text{CTS1}}$
	PA5	I/O	Pull-up Pull-down	o	−	o	TXD1, TB6OUT
	PA6	I/O	Pull-up Pull-down	o	−	o	RXD1, TB6IN
	PA7	I/O	Pull-up Pull-down	o	o	o	TB4IN, INT8
Port B							
	PB0	I/O	Pull-up Pull-down	o	−	o	TRACECLK
	PB1	I/O	Pull-up Pull-down	o	−	o	TRACEDATA0
	PB2	I/O	Pull-up Pull-down	o	−	o	TRACEDATA1
	PB3	I/O	Pull-up Pull-down	o	−	o	TMS/ SWDIO
	PB4	I/O	Pull-up Pull-down	o	−	o	TCK/ SWCLK
	PB5	I/O	Pull-up Pull-down	o	−	o	TDO/ SWV
	PB6	I/O	Pull-up Pull-down	o	−	o	TDI
	PB7	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{TRST}}$
Port C							

Table 8-1 Port Function List

Port	PIn name	Input / Output	Programmable Pull-up Pull-down	Schmitt Input	Noise Filter	Programmable Open-drain	Function pin
	PC0	I/O	Pull-up Pull-down	o	−	o	UO0, SP0DO, SDA0/SO0
	PC1	I/O	Pull-up Pull-down	o	−	o	XO0, SP0DI, SCL0/SI0
	PC2	I/O	Pull-up Pull-down	o	−	o	VO0, SP0CLK, SCK0
	PC3	I/O	Pull-up Pull-down	o	−	o	YO0, SP0FSS
	PC4	I/O	Pull-up Pull-down	o	−	o	WO0, MTOUT00, MTTB0OUT
	PC5	I/O	Pull-up Pull-down	o	−	o	ZO0, MTOUT10, MTTB0IN, SCLK4, $\overline{\text{CTS4}}$
	PC6	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{EMG0}}$, $\overline{\text{GEMG0}}$, TXD4
	PC7	I/O	Pull-up Pull-down	o	−	o	MT0IN, RXD4
Port D							
	PD0	I/O	Pull-up Pull-down	o	o	o	ENCA0, TB5IN, INTC
	PD1	I/O	Pull-up Pull-down	o	−	o	ENCB0, TB5OUT
	PD2	I/O	Pull-up Pull-down	o	o	o	ENCZ0, INTD
	PD3	I/O	Pull-up Pull-down	o	o	o	INT9
	PD4	I/O	Pull-up Pull-down	o	−	o	SCLK2, $\overline{\text{CTS2}}$
	PD5	I/O	Pull-up Pull-down	o	−	o	TXD2
	PD6	I/O	Pull-up Pull-down	o	−	o	RXD2
Port E							
	PE0	I/O	Pull-up Pull-down	o	−	o	TXD0
	PE1	I/O	Pull-up Pull-down	o	−	o	RXD0
	PE2	I/O	Pull-up Pull-down	o	−	o	SCLK0, $\overline{\text{CTS0}}$
	PE3	I/O	Pull-up Pull-down	o	−	o	TB4OUT
	PE4	I/O	Pull-up Pull-down	o	o	o	TB2IN, INT5
	PE5	I/O	Pull-up Pull-down	o	−	o	TB2OUT
	PE6	I/O	Pull-up Pull-down	o	o	o	TB3IN, INT6
	PE7	I/O	Pull-up Pull-down	o	o	o	TB3OUT, INT7
Port F							

Table 8-1 Port Function List

Port	PIn name	Input / Output	Program-mable Pull-up Pull-down	Schmitt Input	Noise Filter	Program-mable Open-drain	Function pin
	PF0	I/O	Pull-up Pull-down	o	－	o	TB7IN
	PF1	I/O	Pull-up Pull-down	o	－	o	TB7OUT, $\overline{ALARM}$
	PF2	I/O	Pull-up Pull-down	o	－	o	ENCA1, SCLK3, $\overline{CTS3}$
	PF3	I/O	Pull-up Pull-down	o	－	o	ENCB1, TXD3
	PF4	I/O	Pull-up Pull-down	o	－	o	ENCZ1, RXD3
Port G							
	PG0	I/O	Pull-up Pull-down	o	－	o	UO1, SDA1 / SO1
	PG1	I/O	Pull-up Pull-down	o	－	o	XO1, SCL1 / SI1
	PG2	I/O	Pull-up Pull-down	o	－	o	VO1, SCK1
	PG3	I/O	Pull-up Pull-down	o	－	o	YO1
	PG4	I/O	Pull-up Pull-down	o	－	o	WO1, MTOUT01, MTTB1OUT
	PG5	I/O	Pull-up Pull-down	o	－	o	ZO1, MTOUT11, MTTB1IN
	PG6	I/O	Pull-up Pull-down	o	－	o	$\overline{EMG1}$, $\overline{GEMG1}$
	PG7	I/O	Pull-up Pull-down	o	－	o	MT1IN
Port H							
	PH0	I/O	Pull-up Pull-down	o	o	o	AIN0, INT0
	PH1	I/O	Pull-up Pull-down	o	o	o	AIN1, INT1
	PH2	I/O	Pull-up Pull-down	o	o	o	AIN2, INT2
	PH3	I/O	Pull-up Pull-down	o	－	o	AIN3
	PH4	I/O	Pull-up Pull-down	o	－	o	AIN4
	PH5	I/O	Pull-up Pull-down	o	－	o	AIN5
	PH6	I/O	Pull-up Pull-down	o	－	o	AIN6
	PH7	I/O	Pull-up Pull-down	o	－	o	AIN7
Port I							
	PI0	I/O	Pull-up Pull-down	o	－	o	AIN8
	PI1	I/O	Pull-up Pull-down	o	－	o	AIN9

Table 8-1 Port Function List

Port	PIn name	Input / Output	Program-mable Pull-up Pull-down	Schmitt Input	Noise Filter	Program-mable Open-drain	Function pin
Port J							
	PJ0	I/O	Pull-up Pull-down	o	−	o	AIN10
	PJ1	I/O	Pull-up Pull-down	o	−	o	AIN11
	PJ2	I/O	Pull-up Pull-down	o	−	o	AIN12
	PJ3	I/O	Pull-up Pull-down	o	−	o	AIN13
	PJ4	I/O	Pull-up Pull-down	o	−	o	AIN14
	PJ5	I/O	Pull-up Pull-down	o	−	o	AIN15
	PJ6	I/O	Pull-up Pull-down	o	o	o	AIN16, INTA
	PJ7	I/O	Pull-up Pull-down	o	o	o	AIN17, INTB
Port L							
	PL0	Output	Pull-up Pull-down	o	−	o	$\overline{\text{BOOT}}$
	PL2	I/O	Pull-up Pull-down	o	o	o	INTF
Port M							
	PM0	I/O	Pull-up Pull-down	o	−	o	X1
	PM1	I/O	Pull-up Pull-down	o	−	o	X2
Port N							
	PN0	I/O	Pull-up Pull-down	o	−	o	SP1DO
	PN1	I/O	Pull-up Pull-down	o	−	o	SP1DI
	PN2	I/O	Pull-up Pull-down	o	−	o	SP1CLK
	PN3	I/O	Pull-up Pull-down	o	−	o	SP1FSS
	PN4	I/O	Pull-up Pull-down	o	−	o	MTOUT02, MTTB2OUT
	PN5	I/O	Pull-up Pull-down	o	−	o	MTOUT12, MTTB2IN
	PN6	I/O	Pull-up Pull-down	o	−	o	$\overline{\text{GEMG2}}$
	PN7	I/O	Pull-up Pull-down	o	o	o	MT2IN, INTE
Port P							
	PP0	I/O	Pull-up Pull-down	o	−	o	XT1
	PP1	I/O	Pull-up Pull-down	o	−	o	XT2

Table 8-1 Port Function List

Port	PIn name	Input / Output	Program-mable Pull-up Pull-down	Schmitt Input	Noise Filter	Program-mable Open-drain	Function pin

o : Exist

- : Not exist

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

8.1.2 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.

8.1.3 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 8-2 shows the pin conditions in STOP mode.

Table 8-2 Port conditions in STOP mode

Function setting	Pin name	I/O	<DRVE> = 0	<DRVE> = 1
Oscillator	X1, XT1	Input	×	×
	X2, XT2	Output	"High" Level Output	"High" Level Output
Port	Px	Input	×	Depend on PxIE[m]
		Output	×	Depend on PxCR[m]
Debug function	TMS/SWDIO TDO/SWV	Input	Depend on PxIE[m]	Depend on PxIE[m]
		Output	Setting with PxCR[m] and it is enabled when data is valid	Setting with PxCR[m] and it is enabled when data is valid
Interrupt function	INT	Input	Depend on PxIE[m]	
SSP	SPCLK, SPFSS, SPDO	Output	×	Setting with PxCR[m] and it is enabled when data is valid
MPT (PMD mode)	$\overline{\text{EMG}}$	Input	×	Depend on PxIE[m]
	UO, VO, WO, XO, YO, ZO	Output	Setting with PxCR[m] and it is enabled when data is valid	Setting with PxCR[m] and it is enabled when data is valid
MPT (IGBT mode)	$\overline{\text{GEMG}}$, MTIN	Input	×	Depend on PxIE[m]
	MTOUT0, MTOUT1	Output	Setting with PxCR[m] and it is enabled when data is valid	Setting with PxCR[m] and it is enabled when data is valid
Other than the above function		Input	×	Depend on PxIE[m]
		Output	×	Depend on PxCR[m]

o : Input or output enabled

× : Input or output disabled

Note:"x" indicates a port number and "m" a corresponding bit.

8.1.4 Precaution on transferring to STOP/SLEEP mode

If PB4 is transferred to the STOP/SLEEP mode while debug function setting of TCK/SWCLK is enabled,it may obtain the less power consumption reduction. When the debug function is not used, set PB4 as a port.

8.1.5 Precaution on exiting STOP mode using interrupts

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

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

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

8.2 Port functions

This chapter describes the port registers detail.

8.2.1 Port A (PA0 to PA7)

The port A is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port A performs the serial interface function (UART / SIO), the 16-bits timer (TMRB), and the external signal interrupt function.

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

8.2.1.1 Port A Register

Base Address = 0x4000_0000

Register name		Address (Base+)
Port A data register	PADATA	0x0000
Port A output control register	PACR	0x0004
Port A function register 1	PAFR1	0x0008
Port A function register 2	PAFR2	0x000C
Port A open drain control register	PAOD	0x0028
Port A pull-up control register	PAPUP	0x002C
Port A pull-down control register	PAPDN	0x0030
Port A input control register	PAIE	0x0038

8.2.1.2 PADATA (Port A data register)

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

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

8.2.1.3 PACR (Port A output control register)

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

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

8.2.1.4 PAFR1 (Port A function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PA7F1	R/W	0: PORT 1: TB4IN
6	PA6F1	R/W	0: PORT 1: RXD1
5	PA5F1	R/W	0: PORT 1: TXD1
4	PA4F1	R/W	0: PORT 1: SCLK1
3	PA3F1	R/W	0: PORT 1: TB1OUT
2	PA2F1	R/W	0: PORT 1: TB1IN
1	PA1F1	R/W	0: PORT 1: TB0OUT
0	PA0F1	R/W	0: PORT 1: TB0IN

8.2.1.5 PAFR2 (Port A function register 2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PA7F2	R/W	0: PORT 1: INT8
6	PA6F2	R/W	0: PORT 1: TB6IN
5	PA5F2	R/W	0: PORT 1: TB6OUT
4	PA4F2	R/W	0: PORT 1: $\overline{\text{CTS1}}$
3	PA3F2	R/W	0: PORT 1: RXIN
2	PA2F2	R/W	0: PORT 1: INT4
1	PA1F2	R/W	0: PORT 1: SCOUT
0	PA0F2	R/W	0: PORT 1: INT3

8.2.1.6 PAOD (Port A open drain control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7OD to PA0OD	R/W	0 : CMOS 1 : Open-drain

8.2.1.7 PAPUP (Port A pull-up control register)

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

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

8.2.1.8 PAPDN (Port A pull-down control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PA7DN to PA0DN	R/W	Pull-Down 0: Disable 1: Enable

8.2.1.9 PAIE (Port A input control register)

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

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

8.2.2 Port B (PB0 to PB7)

The port B is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port B performs the debug communication function and the debug trace output function.

After releasing reset, PB3, PB4, PB5, PB6 and PB7 are functional ports.

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. When PB7 functions as $\overline{\text{TRST}}$ input, pull-up is enabled.

PB0, PB1, PB2 perform as the general-purpose ports with input, output and pull-up disabled.

Note 1: The default setting for PB3 is function port. Input, output, and pull-up are enabled.

Note 2: The default setting for PB4 is function port. Input, and pull-down are enabled.

Note 3: The default setting for PB5 is function port. Output is enabled.

Note 4: The default setting for PB6 and PB7 are function port. Input and pull-up are enabled.

Note 5: If PB3 and PB5 are configured to function port for debug function, outputs are enabled even during Stop mode regardless of the CGSTBYCR<DRVE> bit setting.

Note 6: 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.

8.2.2.1 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 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

8.2.2.2 PBDATA (Port B data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7 to PB0	R/W	Port B data register

8.2.2.3 PBCR (Port B output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
After reset	0	0	1	0	1	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7C to PB0C	R/W	Output 0: Disable 1: Enable

8.2.2.4 PBFR1 (Port B function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7F1	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
After reset	1	1	1	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PB7F1	R/W	0 : PORT 1 : $\overline{\text{TRST}}$
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	PB2F1	R/W	0: PORT 1: TRACEDATA1
1	PB1F1	R/W	0: PORT 1: TRACEDATA0
0	PB0F1	R/W	0: PORT 1: TRACECLK

8.2.2.5 PBOD (Port B open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7OD	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7OD to PB0OD	R/W	0 : CMOS 1 : Open-drain

8.2.2.6 PBPUP (Port B pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7UP	PB6UP	PB5UP	PB4UP	PB3UP	PB2UP	PB1UP	PB0UP
After reset	1	1	0	0	1	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7UP to PB0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.2.7 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	PB7DN	PB6DN	PB5DN	PB4DN	PB3DN	PB2DN	PB1DN	PB0DN
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7DN to PB0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.2.8 PBIE (Port B input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7IE	PB6IE	PB5IE	PB4IE	PB3IE	PB2IE	PB1IE	PB0IE
After reset	1	1	0	1	1	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PB7IE to PB0IE	R/W	Input 0: Disable 1: Enable

8.2.3 Port C (PC0 to PC7)

The port C is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port C performs the Multi-Purpose Timer(16-bits timer, IGBT control, PMD control) (MPT), the synchronous serial interface function (SSP) and the serial bus interface(I2C/SIO).

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

8.2.3.1 Port C Register

Base Address = 0x4000_0080

Register name		Address (Base+)
Port C data register	PCDATA	0x0000
Port C output control register	PCCR	0x0004
Port C function register 1	PCFR1	0x0008
Port C function register 2	PCFR2	0x000C
Port C function register 3	PCFR3	0x0010
Port C function register 4	PCFR4	0x0014
Port C function register 5	PCFR5	0x0018
Port C open drain control register	PCOD	0x0028
Port C pull-up control register	PCPUP	0x002C
Port C pull-down control register	PCPDN	0x0030
Port C input control register	PCIE	0x0038

8.2.3.2 PCDATA (Port C data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PC7 to PC0	R/W	Port C data register

8.2.3.3 PCCR (Port C output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7C	PC6C	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PC7C to PC0C	R/W	Output 0: Disable 1: Enable

8.2.3.4 PCFR1 (Port C function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PC6F1	R/W	0: PORT 1: $\overline{\text{EMG0}}$
5	PC5F1	R/W	0: PORT 1: ZO0
4	PC4F1	R/W	0: PORT 1: WO0
3	PC3F1	R/W	0: PORT 1: YO0
2	PC2F1	R/W	0: PORT 1: VO0
1	PC1F1	R/W	0: PORT 1: XO0
0	PC0F1	R/W	0: PORT 1: UO0

8.2.3.5 PCFR2 (Port C function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7F2	PC6F2	PC5F2	PC4F2	PC3F2	PC2F2	PC1F2	PC0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
7	PC7F2	R/W	0: PORT 1: MT0IN
6	PC6F2	R/W	0: PORT 1: $\overline{\text{GEMG0}}$
5	PC5F2	R/W	0: PORT 1: MTOUT10
4	PC4F2	R/W	0: PORT 1: MTOUT00
3	PC3F2	R/W	0: PORT 1: SP0FSS
2	PC2F2	R/W	0: PORT 1: SP0CLK
1	PC1F2	R/W	0: PORT 1: SP0DI
0	PC0F2	R/W	0: PORT 1: SP0DO

8.2.3.6 PCFR3 (Port C function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	PC5F3	PC4F3	-	PC2F3	PC1F3	PC0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	PC5F3	R/W	0: PORT 1: MTTB0IN
4	PC4F3	R/W	0: PORT 1: MTTB0OUT
3	−	R	Read as 0.
2	PC2F3	R/W	0: PORT 1: SCK0
1	PC1F3	R/W	0: PORT 1: SCL0/SI0
0	PC0F3	R/W	0: PORT 1: SDA0/SO0

8.2.3.7 PCFR4 (Port C function register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7F4	PC6F4	PC5F4	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PC7F4	R/W	0: PORT 1: RXD4
6	PC6F4	R/W	0: PORT 1: TXD4
5	PC5F4	R/W	0: PORT 1: SCLK4
4-0	−	R	Read as 0.

8.2.3.8 PCFR5 (Port C 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	-	-	PC5F5	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	PC5F5	R/W	0: PORT 1: $\overline{\text{CTS4}}$
4-0	−	R	Read as 0.

8.2.3.9 PCOD (Port C open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7OD	PC6OD	PC5OD	PC4OD	PC3OD	PC2OD	PC1OD	PC0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PC7OD to PC0OD	R/W	0 : CMOS 1 : Open-drain

8.2.3.10 PCPUP (Port C pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7UP	PC6UP	PC5UP	PC4UP	PC3UP	PC2UP	PC1UP	PC0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PC7UP to PC0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.3.11 PCPDN (Port C 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	PC7DN	PC6DN	PC5DN	PC4DN	PC3DN	PC2DN	PC1DN	PC0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PC7DN to PC0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.3.12 PCIE (Port C input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE	PC0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PC7IE to PC0IE	R/W	Input 0: Disable 1: Enable

8.2.4 Port D (PD0 to PD6)

The port D 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 D performs the serial interface function (UART/SIO), the 16-bits timer (TMRB), the external signal interrupt function and the encoder input function.

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

8.2.4.1 Port D Register

Base Address = 0x4000_00C0

Register name		Address (Base+)
Port D data register	PDDATA	0x0000
Port D output control register	PDCR	0x0004
Port D function register 1	PDFR1	0x0008
Port D function register 2	PDFR2	0x000C
Port D function register 3	PDFR3	0x0010
Port D open drain control register	PDOD	0x0028
Port D pull-up control register	PDPUP	0x002C
Port D pull-down control register	PDPDN	0x0030
Port D input control register	PDIE	0x0038

8.2.4.2 PDDATA (Port D data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6	PD5	PD4	PD3	PD2	PD1	PD0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PD6 to PD0	R/W	Port D data register

8.2.4.3 PDCR (Port D output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PD6C to PD0C	R/W	Output 0: Disable 1: Enable

8.2.4.4 PDFR1 (Port D function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6F1	PD5F1	PD4F1	PD3F1	PD2F1	PD1F1	PD0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	PD6F1	R/W	0: PORT 1: RXD2
5	PD5F1	R/W	0: PORT 1: TXD2
4	PD4F1	R/W	0: PORT 1: SCLK2
3	PD3F1	R/W	0: PORT 1: INT9
2	PD2F1	R/W	0: PORT 1: ENCZ0
1	PD1F1	R/W	0: PORT 1: ENCB0
0	PD0F1	R/W	0: PORT 1: ENCA0

8.2.4.5 PDFR2 (Port D function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PD4F2	-	-	PD1F2	PD0F2
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4	PD4F2	R/W	0: PORT 1: $\overline{\text{CTS2}}$
3-2	−	R/W	Read as 0.
1	PD1F2	R/W	0: PORT 1: TB5OUT
0	PD0F2	R/W	0: PORT 1: TB5IN

8.2.4.6 PDFR3 (Port D function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PD2F3	-	PD0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PD2F3	R/W	0: PORT 1: INTD
1	−	R	Read as 0.
0	PD0F3	R/W	0: PORT 1: INTC

8.2.4.7 PDOD (Port D open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6OD	PD5OD	PD4OD	PD3OD	PD2OD	PD1OD	PD0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PD6OD to PD0OD	R/W	0 : CMOS 1 : Open-drain

8.2.4.8 PDPUP (Port D pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6UP	PD5UP	PD4UP	PD3UP	PD2UP	PD1UP	PD0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PD6UP to PD0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.4.9 PDPDN (Port D 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	-	PD6DN	PD5DN	PD4DN	PD3DN	PD2DN	PD1DN	PD0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PD6DN to PD0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.4.10 PDIE (Port D input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6IE	PD5IE	PD4IE	PD3IE	PD2IE	PD1IE	PD0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6-0	PD6IE to PD0IE	R/W	Input 0: Disable 1: Enable

8.2.5 Port E (PE0 to PE7)

The port E is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port E performs the serial interface function (UART / SIO), the 16-bits timer function (TMRB), and the external signal interrupt function.

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

8.2.5.1 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 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

8.2.5.2 PEDATA (Port E data register)

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

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

8.2.5.3 PECR (Port E output control register)

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

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

8.2.5.4 PEFR1 (Port E function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7F1	PE6F1	PE5F1	PE4F1	PE3F1	PE2F1	PE1F1	PE0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PE7F1	R/W	0: PORT 1: TB3OUT
6	PE6F1	R/W	0: PORT 1: TB3IN
5	PE5F1	R/W	0: PORT 1: TB2OUT
4	PE4F1	R/W	0: PORT 1: TB2IN
3	PE3F1	R/W	0: PORT 1: TB4OUT
2	PE2F1	R/W	0: PORT 1: SCLK0
1	PE1F1	R/W	0: PORT 1: RXD0
0	PE0F1	R/W	0: PORT 1: TXD0

8.2.5.5 PEFR2 (Port E function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PE7F2	PE6F2	-	PE4F2	-	PE2F2	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PE7F2	R/W	0: PORT 1: INT7
6	PE6F2	R/W	0: PORT 1: INT6
5	−	R	Read as 0.
4	PE4F2	R/W	0: PORT 1: INT5
3	−	R	Read as 0.
2	PE2F2	R/W	0: PORT 1: $\overline{\text{CTS0}}$
1-0	−	R	Read as 0.

8.2.5.6 PEOD (Port E open drain control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PE7OD to PE0OD	R/W	0 : CMOS 1 : Open-drain

8.2.5.7 PEPUP (Port E pull-up control register)

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

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

8.2.5.8 PEPDN (Port E pull-down control register)

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

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

8.2.5.9 PEIE (Port E input control register)

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

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

8.2.6 Port F (PF0 to PF4)

The port F is a general-purpose, 5-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 interface function (UART/SIO) and the 16-bits timer function (TMRB).

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

8.2.6.1 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 3	PFFR3	0x0010
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

8.2.6.2 PFDATA (Port F data register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	PF4 to PF0	R/W	Port F data register

8.2.6.3 PFCR (Port F output control register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	PF4C to PF0C	R/W	Output 0: Disable 1: Enable

8.2.6.4 PFFR1 (Port F function register 1)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4	PF4F1	R/W	0: PORT 1: ENCZ1
3	PF3F1	R/W	0: PORT 1: ENCB1
2	PF2F1	R/W	0: PORT 1: ENCA1
1	PF1F1	R/W	0: PORT 1: TB7OUT
0	PF0F1	R/W	0: PORT 1: TB7IN

8.2.6.5 PFFR2 (Port F function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PF4F2	PF3F2	PF2F2	PF1F2	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4	PF4F2	R/W	0: PORT 1: RXD3
3	PF3F2	R/W	0: PORT 1: TXD3
2	PF2F2	R/W	0: PORT 1: SCLK3
1	PF1F2	R/W	0: PORT 1: $\overline{\text{ALARM}}$
0	−	R	Read as 0.

8.2.6.6 PFFR3 (Port F function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PF2F3	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PF2F3	R/W	0: PORT 1: $\overline{\text{CTS3}}$
1-0	−	R/W	Read as 0.

8.2.6.7 PFOD (Port F open drain control register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	PF4OD to PF0OD	R/W	0 : CMOS 1 : Open-drain

8.2.6.8 PFPUP (Port F pull-up control register)

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

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	PF4UP to PF0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.6.9 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	-	-	-	PE4DN	PE3DN	PE2DN	PE1DN	PE0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	PD4DN to PD0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.6.10 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	-	-	-	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as 0.
4-0	PF4IE to PF0IE	R/W	Input 0: Disable 1: Enable

8.2.7 Port G (PG0 to PG7)

The port G is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port G performs the input/output port for Multi-Purpose Timer(16-bits timer, IGBT control, PMD control) (MPT), the synchronous serial interface (SSP) and the serial bus interface function(I2C/SIO).

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

8.2.7.1 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 function register 3	PGFR3	0x0010
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

8.2.7.2 PGDATA (Port G data register)

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

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

8.2.7.3 PGCR (Port G output control register)

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

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

8.2.7.4 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: $\overline{\text{EMG1}}$
5	PG5F1	R/W	0: PORT 1: ZO1
4	PG4F1	R/W	0: PORT 1: WO1
3	PG3F1	R/W	0: PORT 1: YO1
2	PG2F1	R/W	0: PORT 1: VO1
1	PG1F1	R/W	0: PORT 1: XO1
0	PG0F1	R/W	0: PORT 1: UO1

8.2.7.5 PGFR2 (Port G function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7F2	PG6F2	PG5F2	PG4F2	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PG7F2	R/W	0: PORT 1: MT1IN
6	PG6F2	R/W	0: PORT 1: $\overline{\text{GEMG1}}$
5	PG5F2	R/W	0: PORT 1: MTOUT11
4	PG4F2	R/W	0: PORT 1: MTOUT01
3-0	−	R	Read as 0.

8.2.7.6 PGFR3 (Port G function register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	PG5F3	PG4F3	-	PG2F3	PG1F3	PG0F3
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	−	R	Read as 0.
5	PG5F3	R/W	0: PORT 1: MTTB1IN
4	PG4F3	R/W	0: PORT 1: MTTB1OUT
3	−	R/W	Read as 0.
2	PG2F3	R/W	0: PORT 1: SCK1
1	PG1F3	R/W	0: PORT 1: SI1/SCL1
0	PG0F3	R/W	0: PORT 1: SO1/SDA1

8.2.7.7 PGOD (Port G open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7OD	PG6OD	PG5OD	PG4OD	PG3OD	PG2OD	PG1OD	PG0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7OD to PG0OD	R/W	0 : CMOS 1 : Open-drain

8.2.7.8 PGPUP (Port G pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7UP	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7UP to PG0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.7.9 PGPDN (Port G pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7DN	PG6DN	PG5DN	PG4DN	PG3DN	PG2DN	PG1DN	PG0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7DN to PG0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.7.10 PGIE (Port G input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7IE	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PG7IE to PG0IE	R/W	Input 0: Disable 1: Enable

8.2.8 Port H (PH0 to PH7)

The port H is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port H performs the analog input of the AD converter and the external signal interrupt function.

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

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

8.2.8.1 Port H register

Base Address = 0x4000_01C0

Register name		Address (Base+)
Port H data register	PHDATA	0x0000
Port H output control register	PHCR	0x0004
Port H function register 1	PHFR1	0x0008
Port H open drain control register	PHOD	0x0028
Port H pull-up control register	PHPUP	0x002C
Port H pull-down control register	PHPDN	0x0030
Port H input control register	PHIE	0x0038

8.2.8.2 PHDATA (Port H data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7 to PH0	R/W	Port H data register

8.2.8.3 PHCR (Port H output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7C to PH0C	R/W	Output 0: Disable 1: Enable

8.2.8.4 PHFR1 (Port H function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PH2F1	PH1F1	PH0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PH2F1	R/W	0: PORT 1: INT2
1	PH1F1	R/W	0: PORT 1: INT1
0	PH0F1	R/W	0: PORT 1: INT0

8.2.8.5 PHOD (Port H open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7OD	PH6OD	PH5OD	PH4OD	PH3OD	PH2OD	PH1OD	PH0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7OD to PH0OD	R/W	0 : CMOS 1 : Open-drain

8.2.8.6 PHPUP (Port H pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7UP	PH6UP	PH5UP	PH4UP	PH3UP	PH2UP	PH1UP	PH0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7UP to PH0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.8.7 PHPDN (Port H pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7DN	PH6DN	PH5DN	PH4DN	PH3DN	PH2DN	PH1DN	PH0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7DN to PH0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.8.8 PHIE (Port H input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PH7IE	PH6IE	PH5IE	PH4IE	PH3IE	PH2IE	PH1IE	PH0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PH7IE to PH0IE	R/W	Input 0: Disable 1: Enable

8.2.9 Port I (PI0 to PI1)

The port I 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 port function, the port I performs the analog input of the AD converter.

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

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

8.2.9.1 Port I register

Base Address = 0x4000_0200

Register name		Address (Base+)
Port I data register	PIDATA	0x0000
Port I output control register	PICR	0x0004
Port I open drain control register	PIOD	0x0028
Port I pull-up control register	PIPUP	0x002C
Port I pull-down control register	PIPDN	0x0030
Port I input control register	PIIE	0x0038

8.2.9.2 PIDATA (Port I data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PI1	PI0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PI1 to PI0	R/W	Port I data register

8.2.9.3 PICR (Port I output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PI1C	PI0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PI1C to PI0C	R/W	Output 0: Disable 1: Enable

8.2.9.4 PIOD (Port I open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PI1OD	PI0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PI1OD to PI0OD	R/W	0 : CMOS 1 : Open-drain

8.2.9.5 PIPUP (Port I pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PI1UP	PI0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PI1UP to PI0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.9.6 PIPDN (Port I pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PI1DN	PI0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PI1DN to PI0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.9.7 PIIE (Port I input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PI1IE	PI0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PI1IE to PI0IE	R/W	Input 0: Disable 1: Enable

8.2.10 Port J (PJ0 to PJ7)

The port J is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port J performs the analog input of the AD converter and the external signal interrupt function.

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, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

8.2.10.1 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 function register 1	PJFR1	0x0008
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

8.2.10.2 PJDATA (Port J data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7 to PJ0	R/W	Port J data register

8.2.10.3 PJCR (Port J output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7C to PJ0C	R/W	Output 0: Disable 1: Enable

8.2.10.4 PJFR1 (Port J 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	PJ7F1	PJ6F1	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PJ7F1	R/W	0: PORT 1: INTB
6	PJ6F1	R/W	0: PORT 1: INTA
5-0	−	R	Read as 0.

8.2.10.5 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	PJ5OD	PJ4OD	PJ3OD	PJ2OD	PJ1OD	PJ0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7OD to PJ0OD	R/W	0 : CMOS 1 : Open-drain

8.2.10.6 PJPUP (Port J pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7UP	PJ6UP	PJ5UP	PJ4UP	PJ3UP	PJ2UP	PJ1UP	PJ0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7UP to PJ0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.10.7 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	PJ5DN	PJ4DN	PJ3DN	PJ2DN	PJ1DN	PJ0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7DN to PJ0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.10.8 PJIE (Port J input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7IE	PJ6IE	PJ5IE	PJ4IE	PJ3IE	PJ2IE	PJ1IE	PJ0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7IE to PJ0IE	R/W	Input 0: Disable 1: Enable

8.2.11 Port L (PL0, PL2)

The port L is a general-purpose, 2-bit port. This port has one input port and one input/output port. Besides the general-purpose port, the port L performs the external signal interrupt function and the mode setting function.

While a reset signal is in "Low" state, the PL0(BOOT) input and pull-up are enabled. At the rising edge of the reset signal, if PL0 is "High", the device enters single mode and boots from the on-chip flash memory. If PL0 is "Low", the device enters single boot mode and boots from the internal boot program. For details of single boot mode, refer to "Flash Memory Operation".

Reset initializes the PL0 as general-purpose output port, which output and pull-up are disabled. Reset initializes the PL2 as general-purpose input/output port, which input/output, pull-up and pull-down are disabled.

8.2.11.1 Port L register

Base Address = 0x4000_02C0

Register name		Address (Base+)
Port L data register	PLDATA	0x0000
Port L output control register	PLCR	0x0004
Port L function register 1	PLFR1	0x0008
Port L open drain control register	PLOD	0x0028
Port L pull-up control register	PLPUP	0x002C
Port L pull-down control register	PLPDN	0x0030
Port L input control register	PLIE	0x0038

8.2.11.2 PLDATA (Port L data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PL2	-	PL0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2	R/W	Port L data register
1	−	R	Read as 0.
0	PL0	R/W	Port L data register

8.2.11.3 PLCR (Port L output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PL2C	-	PL0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2C	R/W	Output 0: Disable 1: Enable
1	−	R	Read as 0.
0	PL0C	R/W	Output 0: Disable 1: Enable

8.2.11.4 PLFR1 (Port L 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	-	-	-	-	-	PL2F1	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2F1	R/W	0: PORT 1: INTF
1-0	−	R	Read as 0.

8.2.11.5 PLOD (Port L open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PL2OD	-	PL0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2OD	R/W	0 : CMOS 1 : Open-drain
1	−	R	Read as 0.
0	PL0OD	R/W	0 : CMOS 1 : Open-drain

8.2.11.6 PLPUP (Port L pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PL2UP	-	PL0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2UP	R/W	Pull-up 0: Disable 1: Enable
1	−	R	Read as 0.
0	PL0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.11.7 PLPDN (Port L pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PL2DN	-	PL0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2DN	R/W	Pull-down 0: Disable 1: Enable
1	−	R	Read as 0.
0	PL0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.11.8 PLIE (Port L input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PL2IE	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PL2IE	R/W	Input 0: Disable 1: Enable
1-0	−	R	Read as 0.

8.2.12 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) when CGOSCCR<HOSCON> is "1".

While CGOSCCR<HOSCON> is set to "1",each resister of port M cannot be changed.

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

Note 1: The external high-speed oscillator must not be changed while Port M is "High" level. If you use the external high-speed oscillator, refer to "Clock/Mode Control"

Note 2: After reset, the high-speed clock is selected as the Internal oscillator. Therefore the initial state of port M is a general-purpose port.

8.2.12.1 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

8.2.12.2 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

8.2.12.3 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

8.2.12.4 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

8.2.12.5 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

8.2.12.6 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

8.2.12.7 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

8.2.13 Port N (PN0 to PN7)

The port N is a general-purpose, 8-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port N performs the input/output port for Multi-Purpose Timer(16-bits timer, IGBT control) (MPT), the synchronous serial interface (SSP) and the external signal interrupt function.

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

8.2.13.1 Port N register

Base Address = 0x4000_0340

Register name		Address (Base+)
Port N data register	PNDATA	0x0000
Port N output control register	PNCR	0x0004
Port N function register 1	PNFR1	0x0008
Port N function register 2	PNFR2	0x000C
Port N open drain control register	PNOD	0x0028
Port N pull-up control register	PNPUP	0x002C
Port N pull-down control register	PNPDN	0x0030
Port N input control register	PNIE	0x0038

8.2.13.2 PNDATA (Port N 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	PN7	PN6	PN5	PN4	PN3	PN2	PN1	PN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PN7 to PN0	R/W	Port N data register

8.2.13.3 PNCR (Port N 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	PN7C	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PN7C to PN0C	R/W	Output 0: Disable 1: Enable

8.2.13.4 PNFR1 (Port N 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	PN7F1	PN6F1	PN5F1	PN4F1	PN3F1	PN2F1	PN1F1	PN0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PN7F1	R/W	0: PORT 1: MT2IN
6	PN6F1	R/W	0: PORT 1: $\overline{\text{GEMG2}}$
5	PN5F1	R/W	0: PORT 1:MTOUT12
4	PN4F1	R/W	0: PORT 1:MTOUT02
3	PN3F1	R/W	0: PORT 1: SP1FSS
2	PN2F1	R/W	0: PORT 1: SP1CLK
1	PN1F1	R/W	0: PORT 1: SP1DI
0	PN0F1	R/W	0: PORT 1: SP1DO

8.2.13.5 PNFR2 (Port N 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	PN7F2	-	PN5F2	PN4F2	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	PN7F2	R/W	0: PORT 1: INTE
6	−	R	Read as 0.
5	PN5F2	R/W	0: PORT 1: MTTB2IN
4	PN4F2	R/W	0: PORT 1: MTTB2OUT
3-0	−	R/W	Read as 0.

8.2.13.6 PNOD (Port N 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	PN7OD	PN6OD	PN5OD	PN4OD	PN3OD	PN2OD	PN1OD	PN0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PN7OD to PN0OD	R/W	0 : CMOS 1 : Open-drain

8.2.13.7 PNPUP (Port N 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	PN7UP	PN6UP	PN5UP	PN4UP	PN3UP	PN2UP	PN1UP	PN0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-0	PN7UP to PN0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.13.8 PNPDN (Port N 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	PN7DN	PN6DN	PN5DN	PN4DN	PN3DN	PN2DN	PN1DN	PN0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-0	PN7DN to PN0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.13.9 PNIE (Port N 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	PN7IE	PN6IE	PN5IE	PN4IE	PN3IE	PN2IE	PN1IE	PN0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PN7IE to PN0IE	R/W	Input 0: Disable 1: Enable

8.2.14 Port P (PP0 to PP1)

The port P 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 P performs the external low-speed oscillator (XT1 and XT2).(Note1)

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

Note 1: The external low-speed oscillator must not be changed while Port P is "High" level. If you use the external low-speed oscillator, refer to "Clock/Mode Control".

Note 2: After reset, the external low-speed oscillator is stopped. Therefore the initial state of port P is a general-purpose port .

8.2.14.1 Port P register

Base Address = 0x4000_0380

Register name		Address (Base+)
Port P data register	PPDATA	0x0000
Port P output control register	PPCR	0x0004
Port P open drain control register	PPOD	0x0028
Port P pull-up control register	PPPUP	0x002C
Port P pull-down control register	PPPDN	0x0030
Port P input control register	PPIE	0x0038

8.2.14.2 PPDATA (Port P 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	-	-	-	-	-	-	PP1	PP0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PP1 to PP0	R/W	Port P data register

8.2.14.3 PPCR (Port P 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	-	-	-	-	-	-	PP1C	PP0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PP1C to PP0C	R/W	Output 0: Disable 1: Enable

8.2.14.4 PPOD (Port P 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	-	-	-	-	-	-	PP1OD	PP0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PP1OD to PP0OD	R/W	0 : CMOS 1 : Open-drain

8.2.14.5 PPPUP (Port P 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	-	-	-	-	-	-	PP1UP	PP0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PP1UP to PP0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.14.6 PPPDN (Port P 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	-	-	-	-	-	-	PP1DN	PP0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PP1DN to PP0DN	R/W	Pull-down 0: Disable 1: Enable

8.2.14.7 PPIE (Port P 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	-	-	-	-	-	-	PP1IE	PP0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	PP1IE to PP0IE	R/W	Input 0: Disable 1: Enable

8.3 Block Diagrams of Ports

8.3.1 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 8-3 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

8.3.2 Type FT1

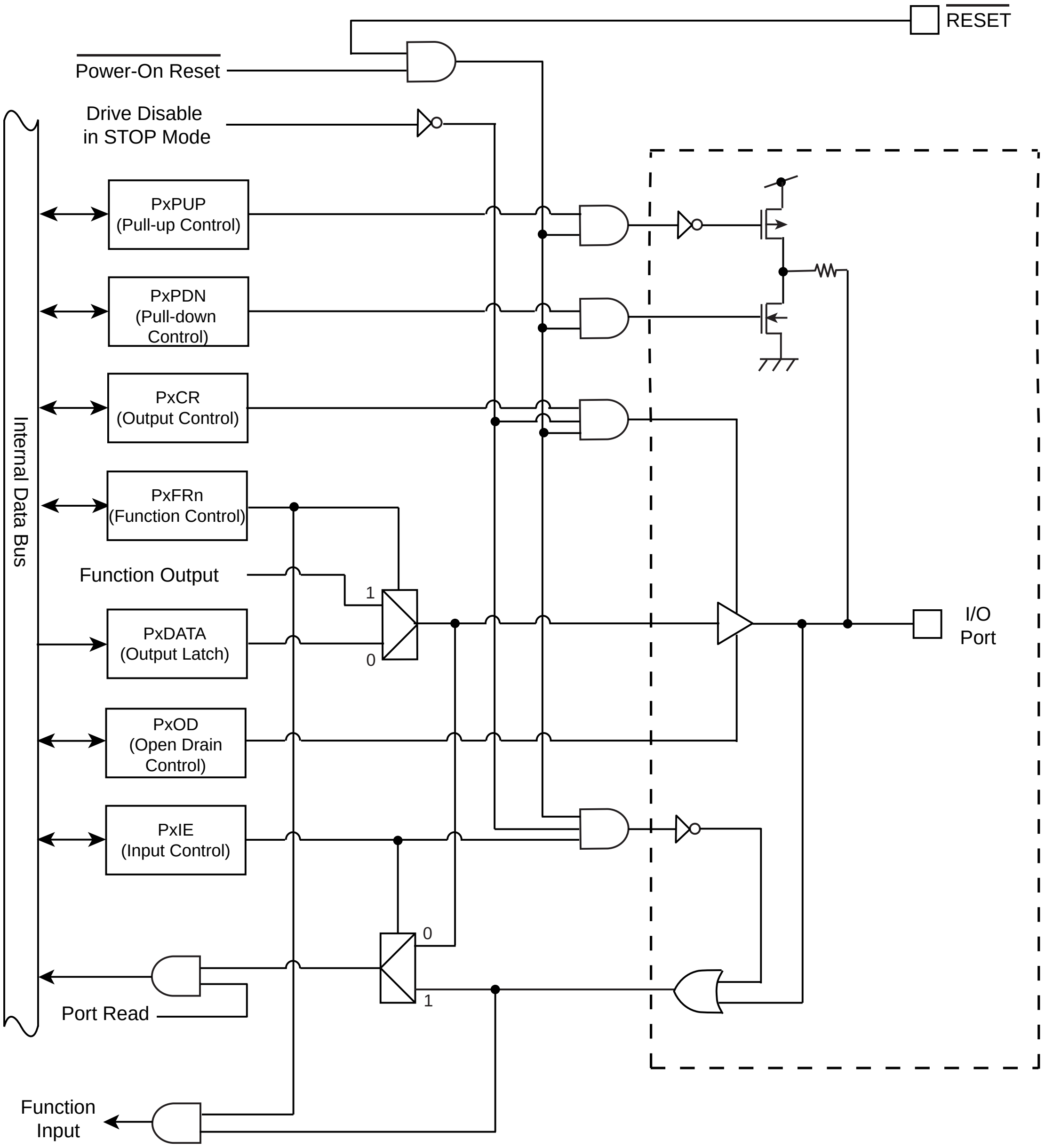


Figure 8-1 Port Type FT1

8.3.3 Type FT2

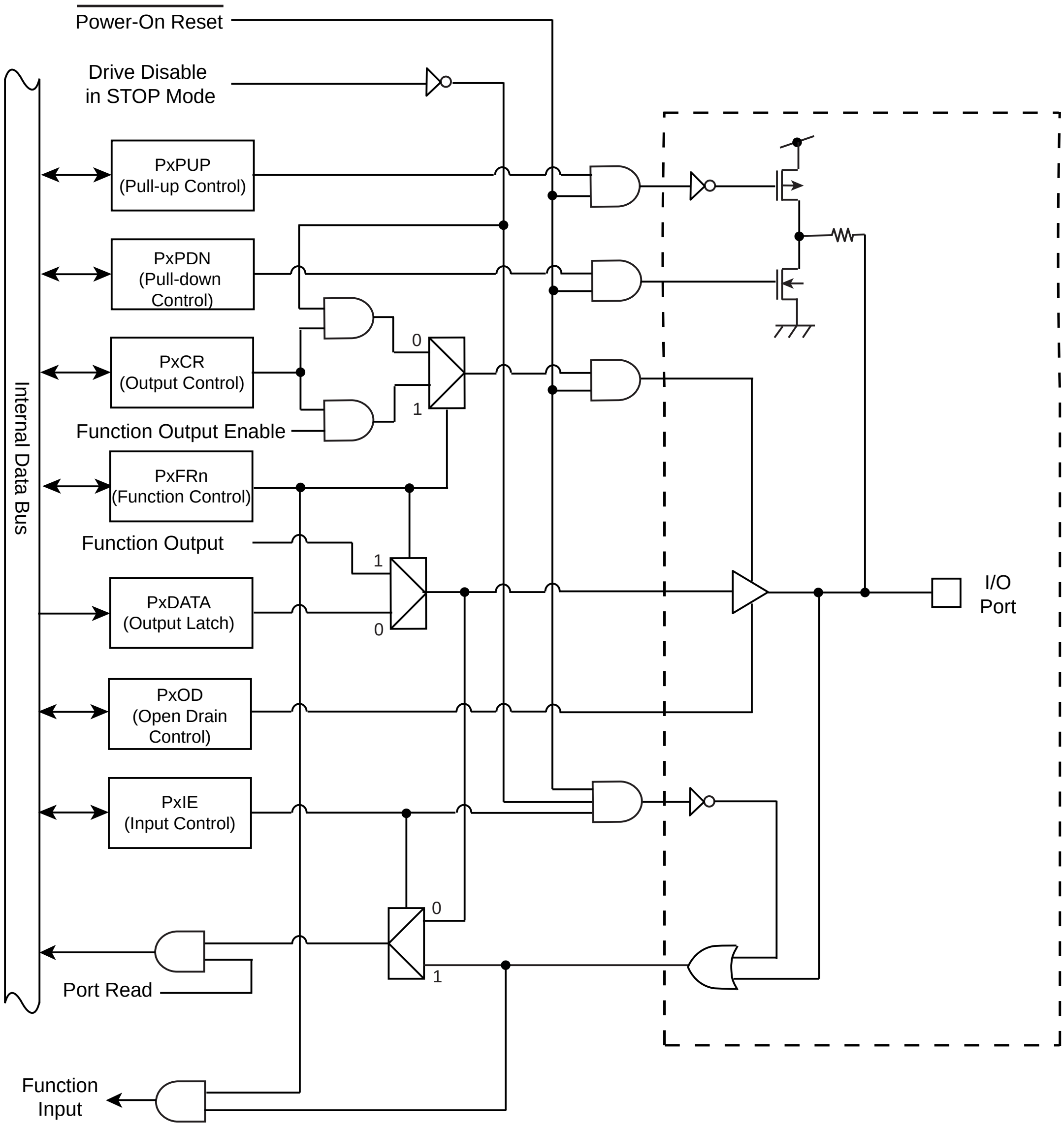


Figure 8-2 Port Type FT2

8.3.4 Type FT3

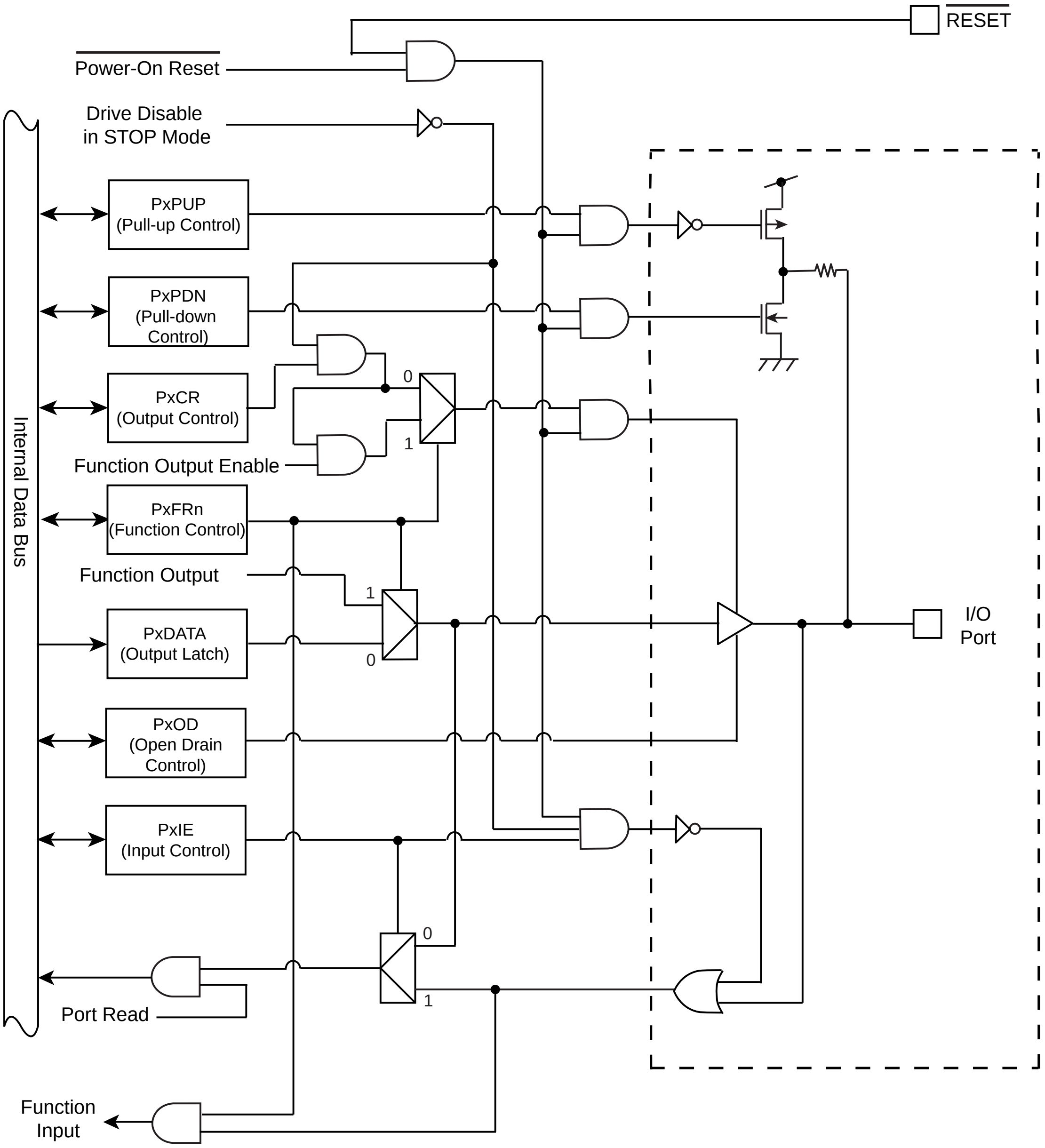


Figure 8-3 Port Type FT3

8.3.5 Type FT4

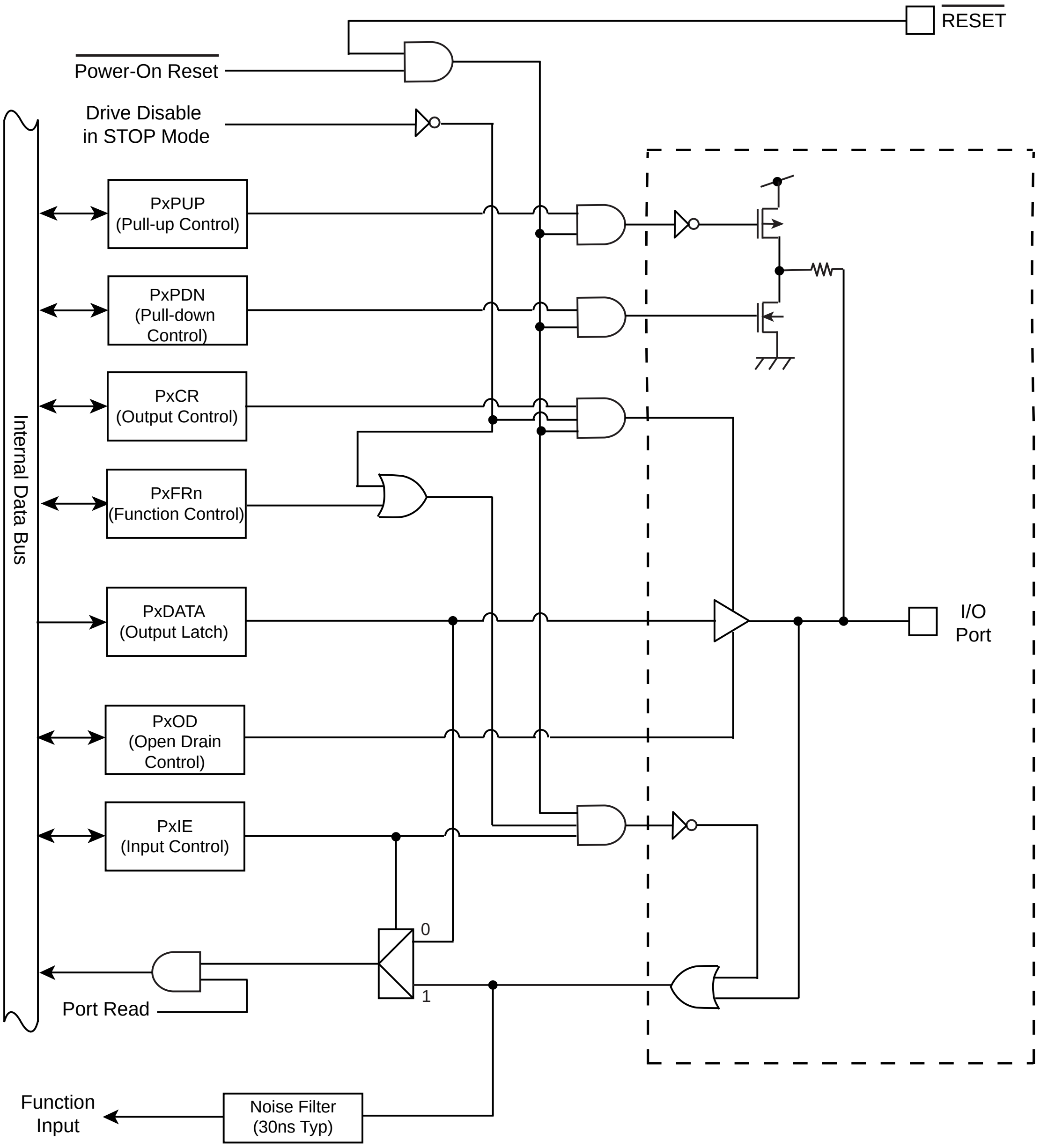


Figure 8-4 Port Type FT4

8.3.6 Type FT5

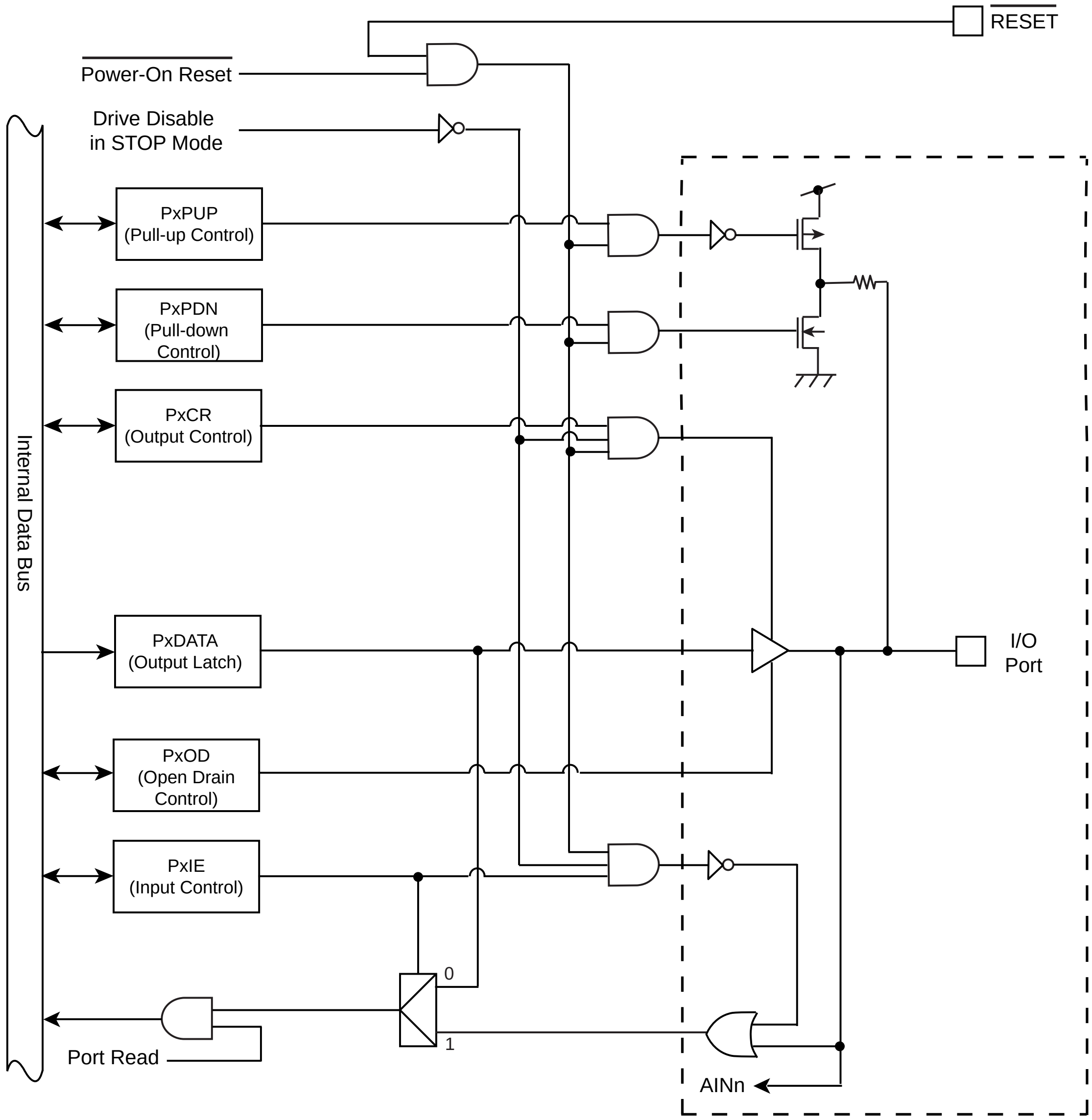


Figure 8-5 Port Type FT5

8.3.7 Type FT6

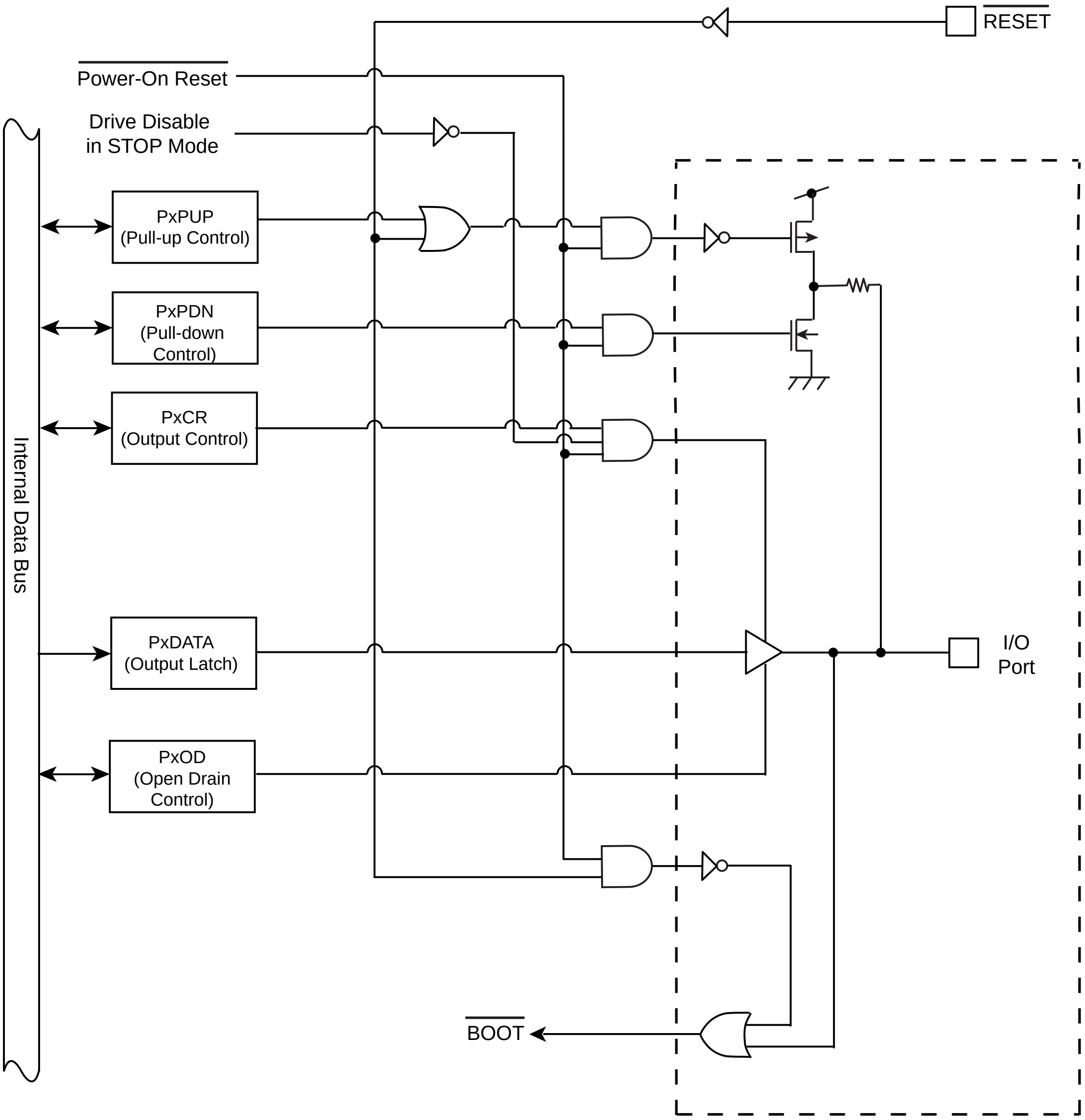


Figure 8-6 Port Type FT6

8.4 Appendix (Port setting List)

The following table shows the register setting for each function.

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

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

8.4.1 Port A setting

Table 8-4 Port Setting List (Port A)

Pin	Port Type	Function	After reset	PACR	PAFR1	PAFR2	PAOD	PAPUP	PAPDN	PAIE
PA0	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	TB0IN (Input)		0	1	0	0	×	×	1
	FT4	INT3 (Input)		0	0	1	0	×	×	1
PA1	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	TB0OUT (Output)		1	1	0	×	×	×	0
		SCOUT (Output)		1	0	1	×	×	×	0
PA2	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	TB1IN (Input)		0	1	0	0	×	×	1
	FT4	INT4 (Input)		0	0	1	0	×	×	1
PA3	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	TB1OUT(Output)		1	1	0	×	×	×	0
		RXIN (Input)		0	0	1	0	×	×	1
PA4	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	SCLK1 (Input)		0	1	0	0	×	×	1
		SCLK1 (Output)		1	1	0	×	×	×	0
		$\overline{\text{CTS1}}$ (Input)		0	0	1	0	×	×	1
PA5	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	TXD1 (Output)		1	1	0	×	×	×	0
		TB6OUT (Output)		1	0	1	×	×	×	0
PA6	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	RXD1 (Input)		0	1	0	0	×	×	1
		TB6IN (Input)		0	0	1	0	×	×	1
PA7	-	Input Port		0	0	0	×	×	×	1
		Output Port		1	0	0	×	×	×	0
	FT1	TB4IN (Input)		0	1	0	0	×	×	1
	FT4	INT8 (Input)		0	0	1	0	×	×	1

8.4.2 Port B Setting

Table 8-5 Port Setting List (Port B)

Pin	Port Type	Function	After reset	PBCR	PBFR1	PBOD	PBPUP	PBPDN	PBIE
PB0	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT1	TRACECLK (Output)		1	1	×	×	×	0
PB1	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT1	TRACEDATA0 (Output)		1	1	×	×	×	0
PB2	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT1	TRACEDATA1 (Output)		1	1	×	×	×	0
PB3	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT2	TMS (Input) / SWDIO (I/O)	o	1	1	0	1	0	1
PB4	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT2	TCK (Input) / SWCLK (Input)	o	0	1	0	0	1	1
PB5	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT2	TDO (Output) / SWV (Output)	o	1	1	0	0	0	0
PB6	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT2	TDI (Input)	o	0	1	0	1	0	0
PB7	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT2	$\overline{\text{TRST}}$ (Input)	o	0	1	0	1	0	1

8.4.3 Port C Setting

Table 8-6 Port Setting List (Port C)

Pin	Port Type	Function	After reset	PC CR	PC FR1	PC FR2	PC FR3	PC FR4	PC FR5	PC OD	PC PUP	PC PDN	PC IE
PC0	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT3	UO0 (Output)		1	1	0	0	0	0	×	×	×	0
		SP0DO (Output)		1	0	1	0	0	0	×	×	×	0
	FT1	SDA0 (I/O)		1	0	0	1	0	0	1	×	×	1
		SO0 (Output)		1	0	0	1	0	0	×	×	×	0
PC1	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT3	XO0 (Output)		1	1	0	0	0	0	×	×	×	0
	FT1	SP0DI (Input)		0	0	1	0	0	0	0	×	×	1
		SCL0 (I/O)		1	0	0	1	0	0	1	×	×	1
		SI0 (Input)		0	0	0	1	0	0	0	×	×	1
PC2	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT3	VO0 (Output)		1	1	0	0	0	0	×	×	×	0
		SP0CLK (Input)		0	0	1	0	0	0	0	×	×	1
		SP0CLK (Output)		1	0	1	0	0	0	×	×	×	0
	FT1	SCK0 (Input)		0	0	0	1	0	0	0	×	×	1
		SCK0 (Output)		1	0	0	1	0	0	×	×	×	0
PC3	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT3	YO0 (Output)		1	1	0	0	0	0	×	×	×	0
		SP0FSS (Input)		0	0	1	0	0	0	0	×	×	1
		SP0FSS (Output)		1	0	1	0	0	0	×	×	×	0
PC4	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT3	WO0 (Output)		1	1	0	0	0	0	×	×	×	0
		MTOUT00 (Output)		1	0	1	0	0	0	×	×	×	0
	FT1	MTTB0OUT (Output)		1	0	0	1	0	0	×	×	×	0
PC5	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT3	ZO0 (Output)		1	1	0	0	0	0	×	×	×	0
		MTOUT10 (Output)		1	0	1	0	0	0	×	×	×	0
	FT1	MTTB0IN (Input)		0	0	0	1	0	0	0	×	×	1
		SCLK4 (Input)		0	0	0	0	1	0	0	×	×	1
		SCLK4 (Output)		1	0	0	0	1	0	×	×	×	0
		$\overline{\text{CTS4}}$ (Input)		0	0	0	0	0	1	0	×	×	1
PC6	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT1	$\overline{\text{EMG0}}$ (Input)		0	1	0	0	0	0	0	×	×	1
		$\overline{\text{GEMG0}}$ (Input)		0	0	1	0	0	0	0	×	×	1
		TXD4 (Output)		1	0	0	0	1	0	×	×	×	0

Table 8-6 Port Setting List (Port C)

Pin	Port Type	Function	After reset	PC CR	PC FR1	PC FR2	PC FR3	PC FR4	PC FR5	PC OD	PC PUP	PC PDN	PC IE
PC7	-	Input port		0	0	0	0	0	0	×	×	×	1
		Output port		1	0	0	0	0	0	×	×	×	0
	FT1	MT0IN (Input)		0	0	1	0	0	0	0	×	×	1
		RXD4 (Input)		0	0	0	0	1	0	0	×	×	1

8.4.4 Port D Setting

Table 8-7 Port Setting List (Port D)

Pin	Port Type	Function	After reset	PD CR	PD FR1	PD FR2	PD FR3	PD OD	PD PUP	PD PDN	PD IE
PD0	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	ENCA0 (Input)		0	1	0	0	0	×	×	1
		TB5IN (Input)		0	0	1	0	0	×	×	1
	FT4	INTC (Input)		0	0	0	1	0	×	×	1
PD1	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	ENCB0 (Input)		0	1	0	0	0	×	×	1
		TB5OUT (Output)		1	0	1	0	×	×	×	0
PD2	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	ENCZ0 (Input)		0	1	0	0	0	×	×	1
	FT4	INTD (Input)		0	0	0	1	0	×	×	1
PD3	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT4	INT9 (Input)		0	1	0	0	0	×	×	1
PD4	-	Input port		0	0	0	−	×	×	×	1
		Output port		1	0	0	−	×	×	×	0
	FT1	SCLK2 (Input)		0	1	0	0	0	×	×	1
		SCLK2 (Output)		1	1	0	0	×	×	×	0
		$\overline{\text{CTS2}}$ (Input)		0	0	1	0	0	×	×	1
PD5	-	Input port		0	0	0	−	×	×	×	1
		Output port		1	0	0	−	×	×	×	0
	FT1	TXD2 (Output)		1	1	0	0	×	×	×	0
PD6	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	RXD2 (Input)		0	0	1	0	0	×	×	1

8.4.5 Port E Setting

Table 8-8 Port Setting List (Port E)

Pin	Port Type	Function	After reset	PECR	PEFR1	PEFR2	PEOD	PEPUP	PEPDN	PEIE
PE0	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	TX0 (Output)		1	1	0	×	×	×	0
PE1	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	RX0 (Input)		0	1	0	0	×	×	1
PE2	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	SCLK0 (Output)		0	1	0	0	×	×	1
		SCLK0 (Output)		1	1	0	×	×	×	0
		$\overline{\text{CTS0}}$ (Input)		0	0	1	0	×	×	1
PE3	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	TB4OUT (Output)		1	1	0	×	×	×	0
PE4	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	TB2IN (Input)		0	1	0	0	×	×	1
	FT4	INT5 (Input)		0	0	1	0	×	×	1
PE5	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	TB2OUT (Output)		1	1	0	×	×	×	0
PE6	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	TB3IN (Input)		0	1	0	0	×	×	1
	FT4	INT6 (Input)		0	0	1	0	×	×	1
PE7	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	TB3OUT (Output)		1	1	0	×	×	×	0
	FT4	INT7 (Input)		0	0	1	0	×	×	1

8.4.6 Port F Setting

Table 8-9 Port Setting List (Port F)

Pin	Port Type	Function	After reset	PF CR	PF FR1	PF FR2	PF FR3	PF OD	PF PUP	PF PDN	PF IE
PF0	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	TB7IN (Input)		0	1	0	0	0	×	×	1
PF1	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	TB7OUT (Output)		1	1	0	0	0	×	×	0
		$\overline{\text{ALARM}}$ (Output)		1	0	1	0	0	×	×	0
PF2	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	ENCA1 (Input)		0	1	0	0	0	×	×	1
		SCLK3 (Input)		0	0	1	0	0	×	×	1
		SCLK3 (Output)		1	0	1	0	×	×	×	0
		$\overline{\text{CTS3}}$ (Input)		0	0	0	1	0	×	×	1
PF3	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	ENCB1 (Input)		0	1	0	0	0	×	×	1
		TXD3 (Output)		1	0	1	0	×	×	×	0
PF4	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	ENCZ1 (Input)		0	1	0	0	0	×	×	1
		RXD3 (Input)		0	0	1	0	0	×	×	1

8.4.7 Port G Setting

Table 8-10 Port Setting List (Port G)

Pin	Port Type	Function	After reset	PG CR	PG FR1	PG FR2	PG FR3	PG OD	PG PUP	PG PDN	PG IE
PG0	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT3	UO1 (Output)		1	1	0	0	×	×	×	0
	FT1	SDA1 (I/O)		1	0	0	1	1	×	×	1
		SO1 (Output)		1	0	0	1	×	×	×	0
PG1	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT3	XO1 (Output)		1	1	0	0	×	×	×	0
	FT1	SCL1 (I/O)		1	0	0	1	1	×	×	1
		SI1 (Input)		0	0	0	1	0	×	×	1
PG2	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT3	VO1 (Output)		1	1	0	0	×	×	×	0
	FT1	SCK1 (Input)		0	0	0	1	0	×	×	1
		SCK1 (Output)		1	0	0	1	×	×	×	0
PG3	--	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT3	YO1 (Output)		1	1	0	0	×	×	×	0
PG4	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT3	WO1 (Output)		1	1	0	0	×	×	×	0
		MTOUT01 (Output)		1	0	1	0	×	×	×	0
	FT1	MTTB1OUT (Output)		1	0	0	1	×	×	×	0
PG5	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT3	ZO1 (Output)		1	1	0	0	×	×	×	0
		MTOUT11 (Output)		1	0	1	0	×	×	×	0
	FT1	MTTB1IN (Input)		0	0	0	1	0	×	×	1
PG6	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	$\overline{\text{EMG1}}$ (Input)		0	1	0	0	0	×	×	1
		$\overline{\text{GEMG1}}$ (Input)		0	0	1	0	0	×	×	1
PG7	-	Input port		0	0	0	0	×	×	×	1
		Output port		1	0	0	0	×	×	×	0
	FT1	MT1IN (Input)		0	0	1	0	0	×	×	1

8.4.8 Port H Setting

Table 8-11 Port Setting List (Port H)

Pin	Port Type	Function	After reset	PHCR	PHFR1	PHOD	PHPUP	PHPDN	PHIE
PH0	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT4	INT0 (Input)		0	1	×	×	×	1
	FT5	AIN0 (Input)		0	0	0	0	0	0
PH1	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT4	INT1 (Input)		0	1	×	×	×	1
	FT5	AIN1 (Input)		0	0	0	0	0	0
PH2	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT4	INT2 (Input)		0	1	×	×	×	1
	FT5	AIN2 (Input)		0	0	0	0	0	0
PH3	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN3 (Input)		0	0	0	0	0	0
PH4	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN4 (Input)		0	0	0	0	0	0
PH5	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN5 (Input)		0	0	0	0	0	0
PH6	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN6 (Input)		0	0	0	0	0	0
PH7	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN7 (Input)		0	0	0	0	0	0

8.4.9 Port I Setting

Table 8-12 Port Setting List (Port I)

Pin	Port Type	Function	After reset	PICR	PIOD	PIPUP	PIPDN	PIIE
PI0	-	Input port		0	×	×	×	1
		Output port		1	×	×	×	0
	FT5	AIN8 (Input)		0	0	0	0	0
PI1	-	Input port		0	×	×	×	1
		Output port		1	×	×	×	0
	FT5	AIN9 (Input)		0	0	0	0	0

8.4.10 Port J Setting

Table 8-13 Port Setting List (Port J)

Pin	Port Type	Function	After reset	PJCR	PJFR1	PJOD	PJPUP	PJPDN	PJIE
PJ0	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN10 (Input)		0	0	0	0	0	0
PJ1	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN11 (Input)		0	0	0	0	0	0
PJ2	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN12 (Input)		0	0	0	0	0	0
PJ3	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN13 (Input)		0	0	0	0	0	0
PJ4	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN14 (Input)		0	0	0	0	0	0
PJ5	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT5	AIN15 (Input)		0	0	0	0	0	0
PJ6	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT4	INTA (Input)		0	1	0	×	×	1
	FT5	AIN16 (Input)		0	0	0	0	0	0
PJ7	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT4	INTB (Input)		0	1	0	×	×	1
	FT5	AIN17 (Input)		0	0	0	0	0	0

8.4.11 Port L Setting

Table 8-14 Port Setting List (Port L)

Pin	Port Type	Function	After reset	PLCR	PLFR1	PLOD	PLPUP	PLPDN	PLIE
PL0	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT6	$\overline{\text{BOOT}}$		×	-	×	×	×	×
PL2	-	Input port		0	0	×	×	×	1
		Output port		1	0	×	×	×	0
	FT4	INTF (Input)		0	1	×	×	×	1

Note: The PL0 input and pull-up are enabled and act as $\overline{\text{BOOT}}$ input pin while a $\overline{\text{RESET}}$ is in "Low" state.

8.4.12 Port M Setting

Table 8-15 Port Setting List (Port M)

Pin	Port Type	Function	After reset	PMCR	PMOD	PMPUP	PMPDN	PMIE
PM0	-	Input port		0	×	×	×	1
		Output port		1	×	×	×	0
	FT5	X1 (Input)		0	0	0	0	0
PM1	-	Input port		0	×	×	×	1
		Output port		1	×	×	×	0
	FT5	X2 (Output)		0	0	0	0	0

8.4.13 Port N setting

Table 8-16 Port Setting List (Port N)

Pin	Port Type	Function	After reset	PNCR	PNFR1	PNFR2	PNOD	PNPUP	PNPDN	PNIE
PN0	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT3	SP1DO (Output)		1	1	0	×	×	×	0
PN1	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	SP1DI (Input)		0	1	0	0	×	×	1
PN2	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT3	SP1CLK (Input)		0	1	0	0	×	×	1
		SP1CLK (Output)		1	1	0	×	×	×	0
PN3	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT3	SP1FSS (Input)		0	1	0	0	×	×	1
		SP1FSS (Output)		1	1	0	×	×	×	0
PN4	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT3	MTOUT02 (Output)		1	1	0	×	×	×	0
	FT1	MTTB2OUT (Output)		1	0	1	×	×	×	0
PN5	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT3	MTOUT12 (Output)		1	1	0	×	×	×	0
	FT1	MTTB2IN (Input)		0	0	1	0	×	×	1
PN6	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	$\overline{\text{GEMG2}}$ (Input)		0	1	0	0	×	×	1
PN7	-	Input port		0	0	0	×	×	×	1
		Output port		1	0	0	×	×	×	0
	FT1	MT2IN (Input)		0	1	0	0	×	×	1
	FT4	INTE (Input)		0	0	1	0	×	×	1

8.4.14 Port P Setting

Table 8-17 Port Setting List (Port P)

Pin	Port Type	Function	After reset	PPCR	PPOD	PPPUP	PPPDN	PPIE
PP0	-	Input port		0	×	×	×	1
		Output port		1	×	×	×	0
	FT5	XT1 (Input)		0	0	0	0	0
PP1	-	Input port		0	×	×	×	1
		Output port		1	×	×	×	0
	FT5	XT2 (Output)		0	0	0	0	0

9. 16-bit Timer / Event Counters (TMRB)

9.1 Outline

TMRB operate in the following four operation modes:

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

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

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

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

9.2 Differences in the Specifications

TMPM3U6FY/FW contains 8-channel of TMRB.

Each channel functions independently and the channels operate in the same way except for the differences in their specification as shown in Table 9-1.

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

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

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

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

Table 9-1 Differences in the Specifications of TMRB Modules

Specification	External pins		Trigger function between timers		Interrupt		Internal Connects	
Channel	External clock / capture trigger input pins	Timer Flip-Flop output pins	Capture trigger	Synchronous start trigger channel	Capture interrupt	TMRB interrupt	Start AD conversion	Timer Flip-Flop Connect with SIO/UART, RMC (TXTRG : Transfer clock)
TMRB0	TB0IN	TB0OUT	TB7OUT	-	INTCAP00 INTCAP01	INTTB00 INTTB01	-	-
TMRB1	TB1IN	TB1OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP10 INTCAP11	INTTB10 INTTB11	-	RMC
TMRB2	TB2IN	TB2OUT	TB7OUT	TB0PRUN TB0RUN	INTCAP20 INTCAP21	INTTB20 INTTB21	-	-
TMRB3	TB3IN	TB3OUT	TB2OUT	TB0PRUN TB0RUN	INTCAP30 INTCAP31	INTTB30 INTTB31	-	-
TMRB4	TB4IN	TB4OUT	TB2OUT	-	INTCAP40 INTCAP41	INTTB40 INTTB41	-	SIO0 SIO1
TMRB5	TB5IN	TB5OUT	TB2OUT	TB4PRUN TB4RUN	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51	-
TMRB6	TB6IN	TB6OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP60 INTCAP61	INTTB60 INTTB61	-	-
TMRB7	TB7IN	TB7OUT	TB5OUT	TB4PRUN TB4RUN	INTCAP70 INTCAP71	INTTB70 INTTB71	-	SIO2 SIO3

9.3 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.

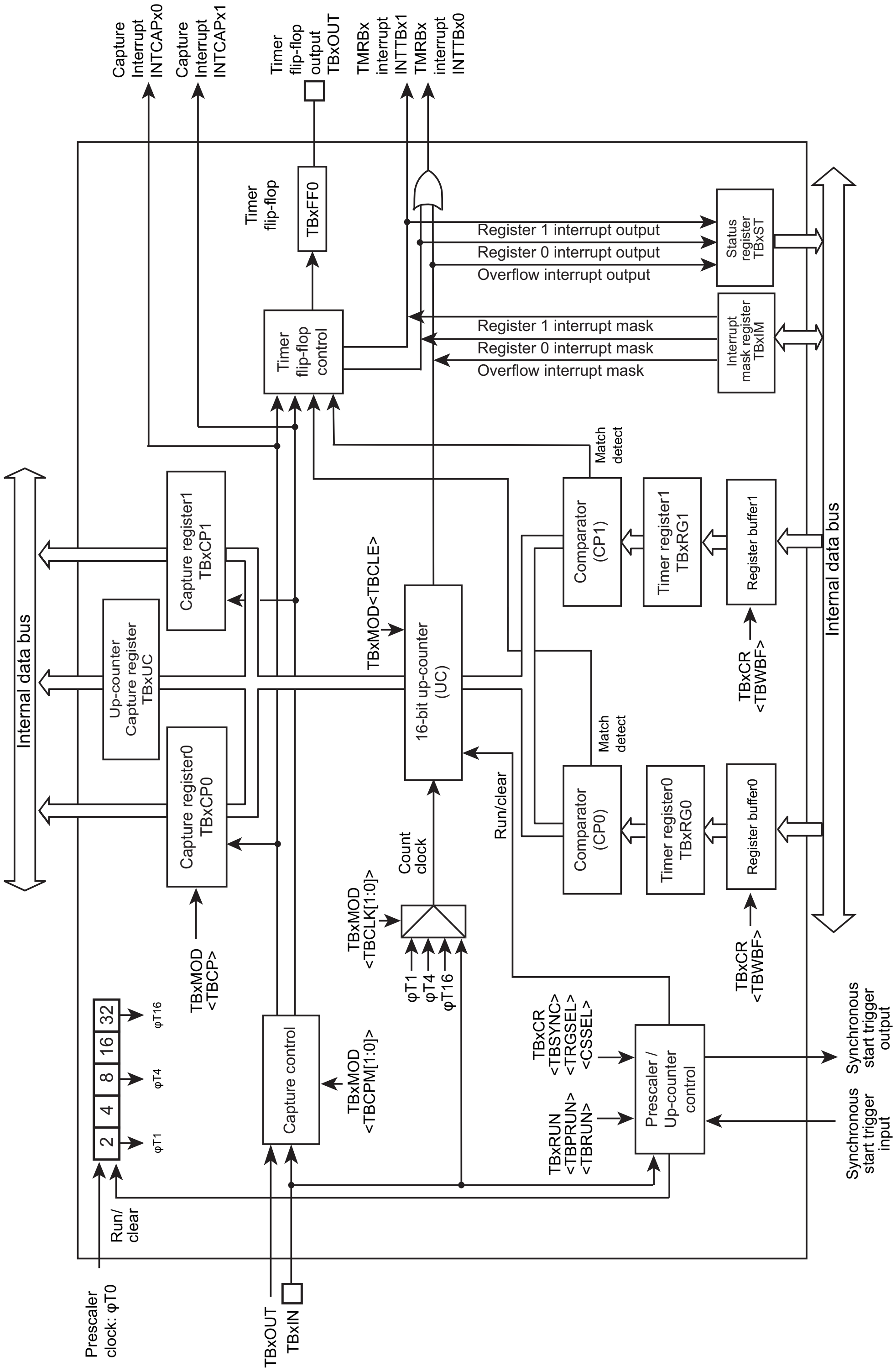


Figure 9-1 TMRBx Block Diagram

9.4 Registers

9.4.1 Register list according to channel

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

Channel x	**Base Address**
Channel 0	0x4001_0000
Channel 1	0x4001_0040
Channel 2	0x4001_0080
Channel 3	0x4001_00C0
Channel 4	0x4001_0100
Channel 5	0x4001_0140
Channel 6	0x4001_0180
Channel 7	0x4001_01C0

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

9.4.2 TBxEN (Enable register)

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

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

9.4.3 TBxRUN (RUN register)

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

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

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

9.4.4 TBxCR (Control register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0"
7	TBWBF	R/W	Double buffer 0: Disable 1: Enable
6	−	R/W	Write as "0".
5	TBSYNC	R/W	Synchronous mode switching 0: individual (unit of channel) 1: synchronous
4	−	R	Read as "0".
3	I2TB	R/W	Operation at IDLE mode 0: Stop 1:Operation
2	−	R	Read as "0"
1	TRGSEL	R/W	Select external trigger 0: Rising edge 1: Falling edge Select the edge of the external trigger (Signal to TBxIN pin)
0	CSSEL	R/W	Select count start 0: Soft start 1: External trigger

9.4.5 TBxMOD (Mode register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6	TBRSWR	R/W	Controls the timing to write to timer registers 0 and 1 when double buffering is enabled. 0: Timer registers 0 and 1 can be written separately, even in case writing preparation is ready for only one resister. 1: In case both resisters are not ready to be written, Timer registers 0 and 1 can not be written
5	TBCP	W	Capture control by software 0: Capture by software 1: Don’t care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as "1".
4-3	TBCPM[1:0]	R/W	Capture timing 00: Disable 01: TBxIN↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. 10: TBxIN↑ TBxIN↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN pin input. 11: TBxOUT↑ TBxOUT↓ Takes count values into capture register 0 (TBnCP0) upon rising of 16-bit timer match output (TBxOUT) and into capture register 1 (TBnCP1) upon falling of TBxOUT. (x=7, n=0,1,2), (x=2, n=3,4,5), (x=5, n=6,7), (TMRB0 to 2: TB7OUT, TMRB3 to 5: TB2OUT, TMRB6 to 7: TB5OUT)
2	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter. 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up counter when there is a match with Timer Regsiter1 (TBxRG1).
1-0	TBCLK[1:0]	R/W	Selects the TMRBx source clock. 00: TBxIN pin input 01: φT1 10: φT4 11: φT16

Note: Do not modify TBxMOD during operating TMRBx.

9.4.6 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".

9.4.7 TBxST (Status register)

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

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

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

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

9.4.8 TBxIM (Interrupt mask register)

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

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

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

9.4.9 TBxUC (Up counter capture register)

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

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

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

9.4.10 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.

9.4.11 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.

9.4.12 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.

9.4.13 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.

9.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 9-1.

9.5.1 Prescaler

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

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

The operation or the stoppage of a prescaler is set with TBxRUN<TBPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Below tables show prescaler output clock resolutions.

Table 9-2 Prescaler Output Clock Resolutions (fc = 40MHz)

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

Table 9-2 Prescaler Output Clock Resolutions (fc = 40MHz)

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

Note 1: The prescaler output clock φTn must be selected so that φTn < fsys is satisfied (so that φTn is slower than fsys).

Note 2: Do not change the clock gear while the timer is operating.

Note 3: "." denotes a setting prohibited.

Table 9-3 Prescaler Output Clock Resolutions (fs = 32.768kHz, <SYSCK> = "1")

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

9.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

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

- Counter start / stop

  Counter operation is specified by TBxRUN<TBRUN>. UC starts counting if <TBRUN> = "1", and stops counting and clears counter value if <TBRUN> = "0".

- Timing to clear UC

  1. When a match is detected.

     By setting TBxMOD<TBCLE> = "1", UC is cleared if when the comparator detects a match between counter value and the value set in TBxRG1. UC operates as a free-running counter if TBxMOD<TBCLE> = "0".

  2. When UC stops

     UC stops counting and clears counter value if TBxRUN<TBRUN> = "0".

- UC overflow

  If UC overflow occurs, the INTTBx overflow interrupt is generated.

9.5.3 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.

9.5.4 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>.

9.5.5 Capture registers (TBxCP0, TBxCP1)

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

9.5.6 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.

9.5.7 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.

9.5.8 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.

9.5.9 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.

9.6 Description of Operations for Each Mode

9.6.1 16-bit interval Timer Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger
TBxMOD	←	X	0	1	0	0	1	*	*	Changes to prescaler output clock as input clock. Specifies capture function to disable.
						(*** = 01, 10, 11)				
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don’t care −; No change

9.6.2 16-bit Event Counter Mode

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

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

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

Note: X; Don’t care −; No change

9.6.3 16-bit PPG (Programmable Pulse Generation) Output Mode

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

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

Set value of TBxRG0 < Set value of TBxRG1

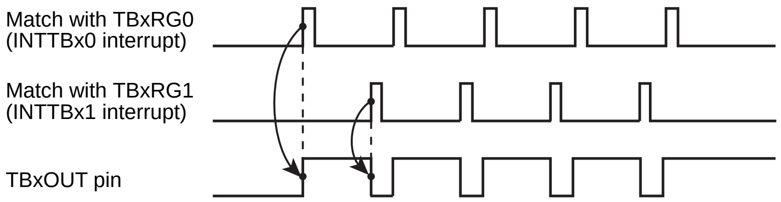


Figure 9-2 Example of Output of Programmable Pulse Generation (PPG)

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.

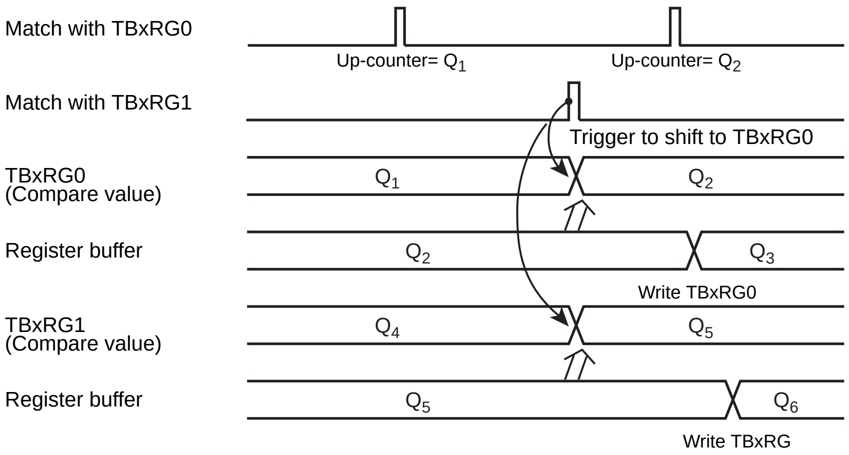


Figure 9-3 Register Buffer Operation

The block diagram of this mode is shown below.

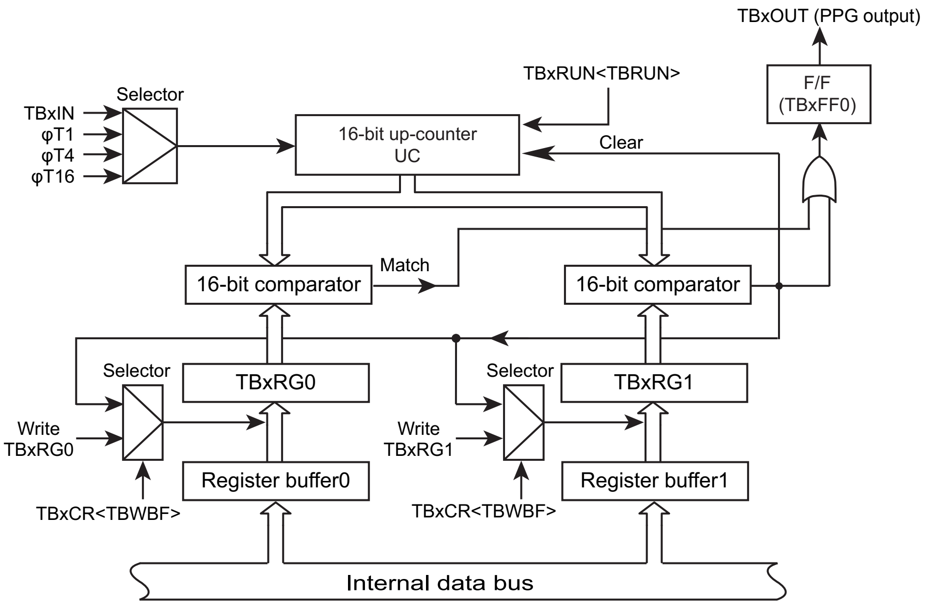


Figure 9-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	−	X	−	X	0	0	Disable double buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits))
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	X	0	0	0	0	0	Enables the TBxRG0 / TBxRG1 double buffering. (Changes the duty / cycle when the INTTBx interrupt is generated)
TBxFFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TBxFF0 to reverse when a match with TBxRG0 or TBxRG1 is detected, and sets the initial value of TBxFF0 to "0".
TBxMOD	←	X	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function.
						(*** = 01, 10, 11)				
Set PORT registers.										Allocates corresponding port to TBxOUT.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note 1: m ; corresponding bit of port
Note 2: X; Don't care
−; No change

9.6.4 External Trigger PPG (Programmable Square Wave) Output Mode

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

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

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

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

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

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

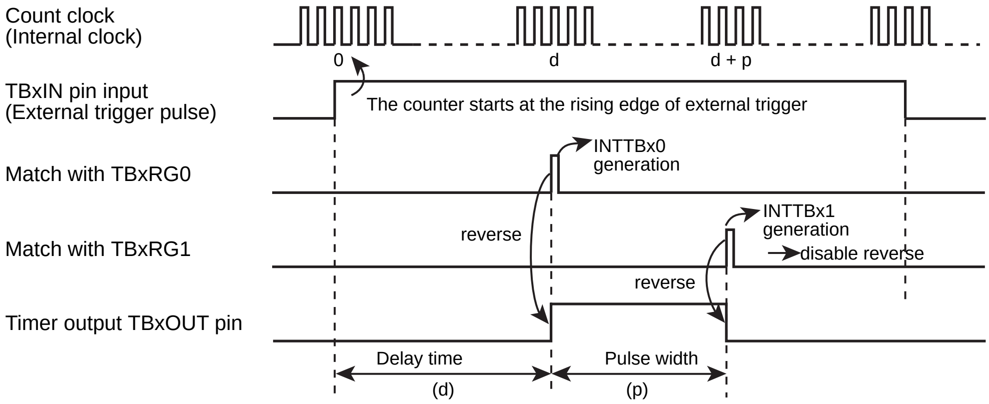


Figure 9-5 One-shot pulse generation using an external trigger count start (with a delay)

9.6.5 Timer synchronous mode

This mode enables the timers to start synchronously.

If the mode is used with PPG output, the output can be applied to drive a motor.

TMRB is consisted of pairs of 4-channel TMRB. If one channel starts, remaining 3 channels can be start synchronously. In the TMPM3U6FY/FW, the following combinations allow to use.

Start trigger channel (Master channel)	Synchronous operation channel (Slave channel)
TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6, TMRB7

Use of the timer synchronous mode is specified in TBxCR<TBSYNC> bit.

- <TBSYNC> = "0" : Timer operates individually.
- <TBSYNC> = "1" : Timers operates synchronously.

Set "0" to the <TBSYNC> bit in the master channel.

If <TBSYNC>= "1" is set in the slave channel, the start timing is synchronized with master channel start timing. Setting of start timing for TBxRUN<TBPRUN, TBRUN> bit in the slave channel is not required.

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

9.7 Applications using the Capture Function

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

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

9.7.1 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 UC matches TBxRG0 and TBxRG1. This trigger is disabled by the INTTBx interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in "Figure 9-6 One-shot Pulse Output (With Delay)".

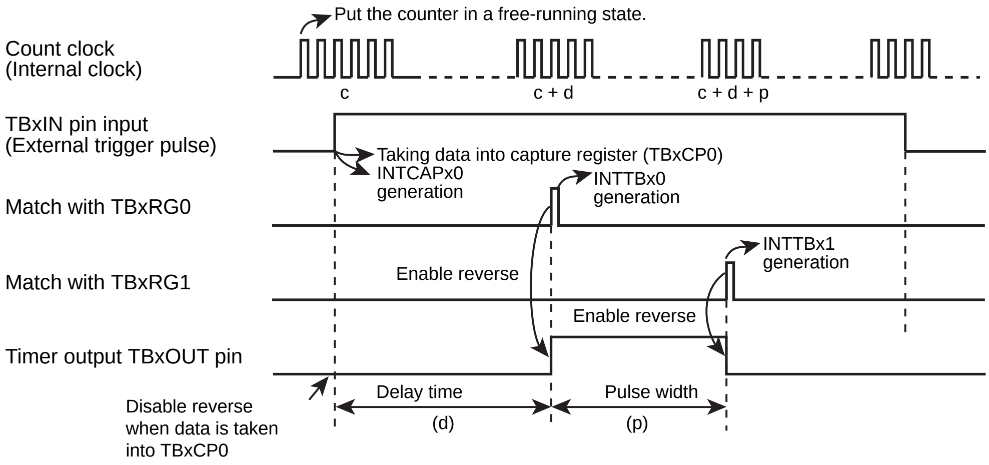


Figure 9-6 One-shot Pulse Output (With Delay)

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	←	X	1	0	1	0	0	0	1	Changes source clock to φT1. Fetches a count value into the TBxCP0 at the rising edge of 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 INTTBx interrupt corresponding bit setting to "1".
[Processing of INTTBx interrupt service routine] Output disable										
TBxFFCR	←	X	X	−	−	0	0	−	−	Clears TBxFF0 reverse trigger setting.
Interrupt enable clear register	←	*	*	*	*	*	*	*	*	Prohibits interrupts specified by INTTBx interrupt corresponding bit by setting to "1".

Note 1: m ; corresponding bit of port

Note 2: X; Don't care

−; No change

If a delay is not required, TBxFF0 is reversed when data is taken into TBxCP0, and TBxRG1 is set to the sum of the TBxCP0 value (c) and the one-shot pulse width (p), (c + p), by generating the INTCAPx0 interrupt. TBxRG1 change must be completed before the next match.

TBxFF0 is enabled to reverse when UC matches with TBxRG1, and is disabled by generating the INTTBx interrupt.

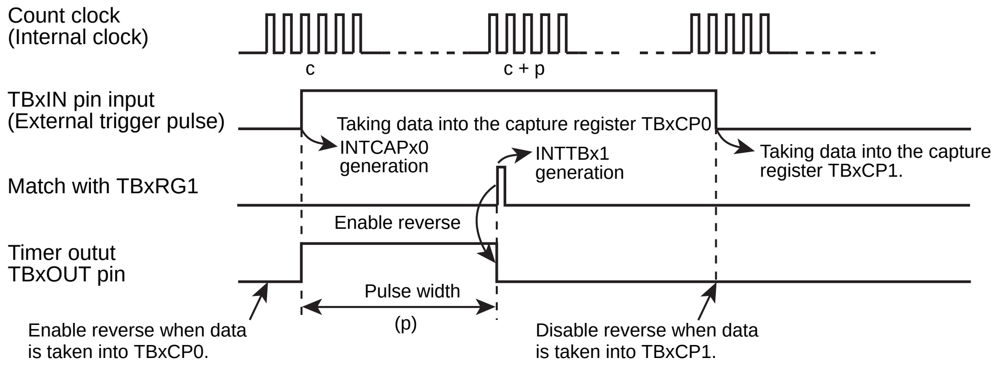


Figure 9-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

9.7.2 Frequency measurement

The frequency of an external clock can be measured by using the capture function.

To measure frequency, another 16-bit timer is used in combination with the 16-bit event counter mode. As an example, we explain with TMRB0 and TMRB7. TB7OUT of the 16-bit timer TMRB7 is used to specify the measurement time.

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

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

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

For example, if the difference between TB0CP0 and TB0CP1 is 100 and the level width setting value of TB7OUT is 0.5 s, the frequency is 200 Hz (100 ÷ 0.5 s = 200 Hz).

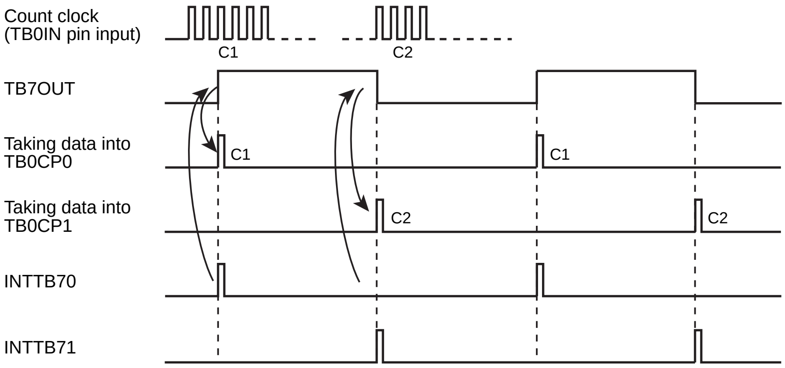


Figure 9-8 Frequency Measurement

9.7.3 Pulse width measurement

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

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

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

Caution must be exercised when measuring pulse widths exceeding the UC maximum count time which is dependent upon the source clock used. The measurement of such pulse widths must be made using software.

The "Low" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAPx0 interrupt processing as shown in "Figure 9-9 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

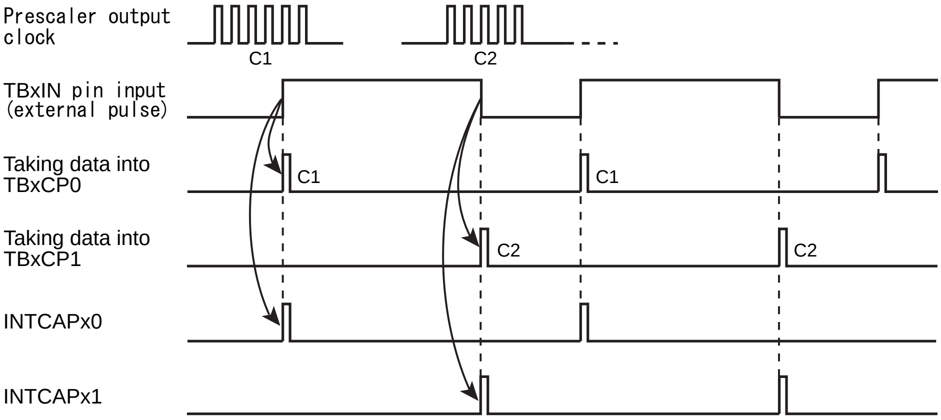


Figure 9-9 Pulse Width Measurement

10. 16-bit Multi-Purpose Timer (MPT)

10.1 Outline

TMPM3U6FY/FW has 3 channels of 16-bit multi-purpose timer

(MPT). The MPT provides three operational modes as follows.

<Timer mode>

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable rectangular waveform output (PPG, one output) mode
- Pulse width measurement (capture)

<IGBT mode>

- 16-bit programmable rectangular waveform output (PPG, two outputs) mode
- External trigger start
- Cycle match detection
- Emergency stop function

<PMD mode>

- 3-phase motor control mode

Hereafter, "x" indicates a channel number.

Note: **MPT2 do not have PMD mode.**

10.2 Specification Differences in Channel line-up

Each channel (MPT0-MPT2) operates independently and identically except the differences shown in the Table 10-1.

Table 10-1 MPT specification differences in channel line-up

Specification Channel	External pin					
	External clock/ capture trigger input pin	Timer flip-flop output pin	IGBT input pin	IGBT output pin	PMD input pin	PMD output pin
MPT 0	MTTB0IN	MTTB0OUT	$\overline{\text{GEMG0}}$ MT0IN	MTOUT00 MTOUT10	$\overline{\text{EMG0}}$	UO0,VO0, WO0,XO0, YO0,ZO0
MPT 1	MTTB1IN	MTTB1OUT	$\overline{\text{GEMG1}}$ MT1IN	MTOUT01 MTOUT11	$\overline{\text{EMG1}}$	UO1,VO1, WO1,XO1, YO1,ZO1
MPT 2	MTTB2IN	MTTB2OUT	$\overline{\text{GEMG2}}$ MT2IN	MTOUT02 MTOUT12	-	-

10.3 Structure

The MPT consists of three modules including a timer, IGBT and PMD. Each module is switched by registers.

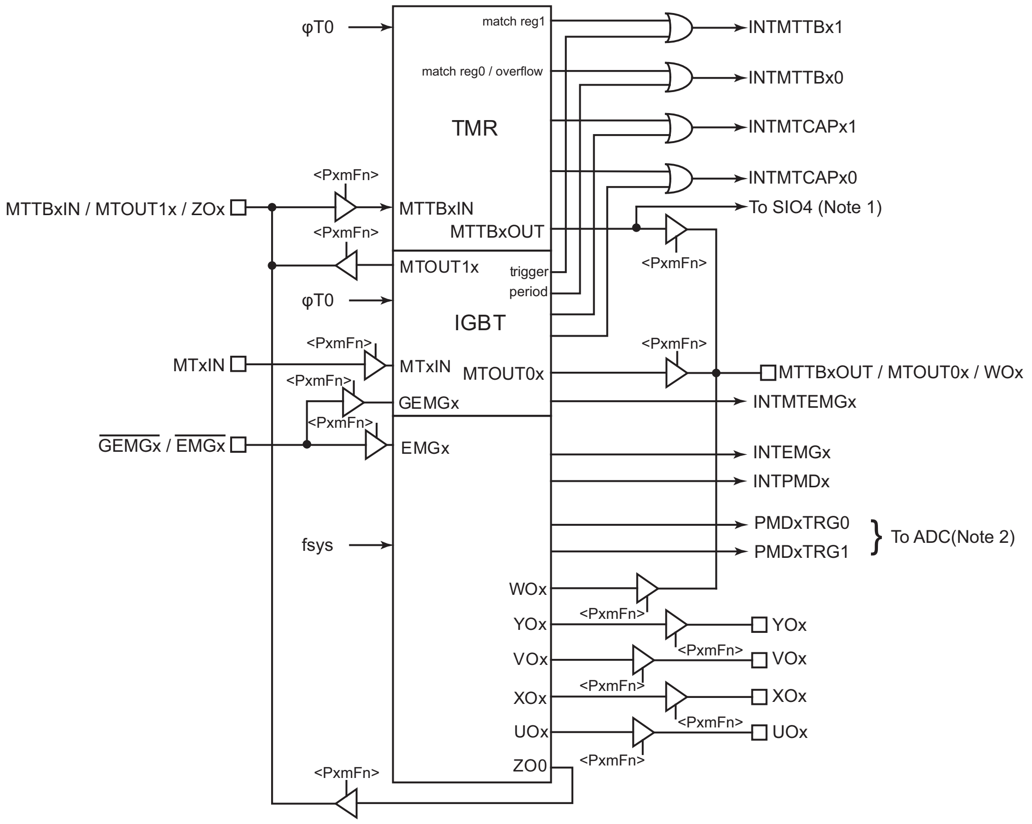


Figure 10-1 Block Diagram of MPTx

Note 1: **Timer flip-flop output (MTTB0OUT) in the MPT0 timer mode can be used as a serial transfer clock in the SIO4 UART mode.**

Note 2: MPT2 do not have PMD module.

10.4 Operation Description of Timer Mode

10.4.1 Block Diagram

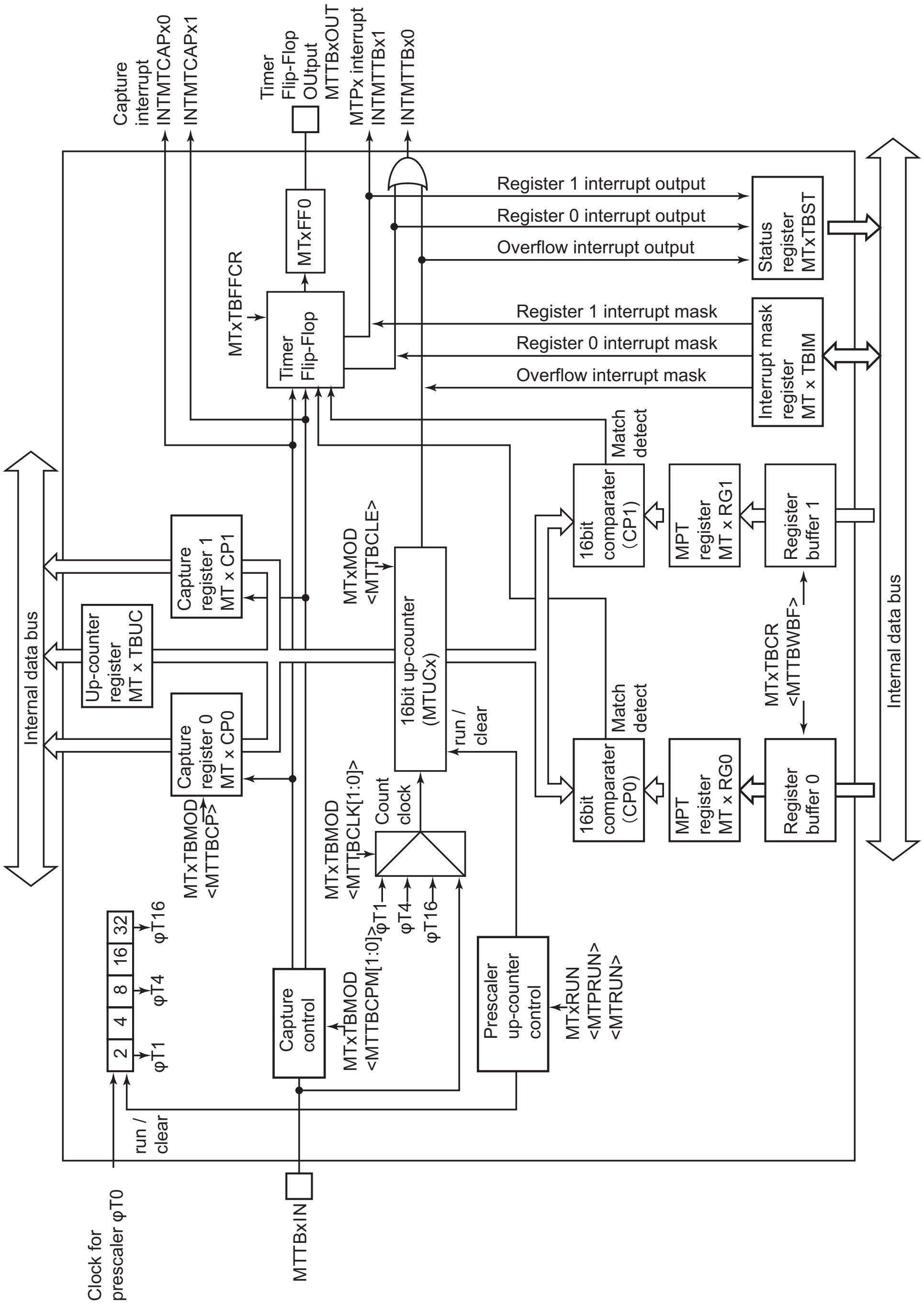


Figure 10-2 Block Diagram of Timer Mode

10.4.2 Registers categorized by timer mode channel

This section describes registers and addresses of each channel.

Channel x	Base Address
MPT 0	0x4005_0800
MPT 1	0x4005_0880
MPT 2	0x4005_0900

Register name (x = 0 to 2)		Address (Base+)
MPT enable register	MTxEN	0x0000
MPT RUN register	MTxRUN	0x0004
MPT control register	MTxTBCR	0x0008
MPT mode register	MTxTBMOD	0x000C
MPT flip-flop control register	MTxTBFFCR	0x0010
MPT status register	MTxTBST	0x0014
MPT interrupt mask register	MTxTBIM	0x0018
MPT up-counter register	MTxTBUC	0x001C
MPT register	MTxRG0 MTxRG1	0x0020 0x0024
MPT capture register	MTxCP0 MTxCP1	0x0028 0x002C

10.4.3 MTxEN (MPT enable register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	MTEN	R/W	Specifies MPT operation. 0: Disable 1: Enable When MTEN is disabled, feeding clock to other registers of MPT module is stopped, so that power consumption can be reduced. (Read or write to other registers cannot be done.)
6	MTHALT	R/W	Specifies MPT operation when core halts (debug break). [TMR function] 0: Clock stopping operation is disabled while core halts. 1: Clock stopping operation is enabled while core halts. [IGBT function] 0: Not control clock stopping operation and MTOUT0x/MTOUT1x output. 1: Clock stopping operation is enabled while core halt. It controls MTOUT0x/MTOUT1x output according to the MTxIGEMGCR<IGEMGOC> setting.
5-1	-	R	Read as "0".
0	MTMODE	R/W	Specifies operation modes 0: Timer mode 1: IGBT mode

Note: **When MPT is used, MPT operation is enabled (<MTEN> = "1") before each register of MPT module is set. Even if MPT operation is disabled after MPT is stopped, each register setting is maintained.**

10.4.4 MTxRUN (MPT RUN register)

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

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTPRUN	R/W	Controls MPT prescaler operation 0: Stops prescaler operation. Prescaler is cleared to "0". 1: Starts prescaler operation.
1	-	R	Read as "0".
0	MTRUN	R/W	Controls MPT counting operation 0: Stops counting operation. Counter is cleared to "0". 1: Starts counting operation.

10.4.5 MTxTBCR (MPT control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTTBWBF	-	-	-	MTI2TB	-	MTTB TRGSEL	MTTBCSSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	MTTBWBF	R/W	Specifies double buffer to enable/disable 0:Disabled 1:Enabled
6-5	-	R/W	Write "0".
4	-	R	Read as "0".
3	MTI2TB	R/W	Controls clock operation to start/stop in IDLE mode 0:Stop 1:Start
2	-	R	Read as "0".
1	MTTBTRGSEL	R/W	Selects rising or falling edge of external trigger. 0:Rising edge 1:Falling edge
0	MTTBCSSEL	R/W	Selects how to start counting 0:Soft start 1:External trigger

Note 1: **Do not modify MTxTBCR during timer in operation (MTxRUN<MTRUN> = "1").**

Note 2: **In the IGBT mode, double-buffering is automatically enabled regardless of <MTTBWBF> setting.**

10.4.6 MTxTBMOD (MPT mode register)

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

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as "0".
6	MTTBRSWR	R/W	Controls the write timing to timer register 0 and 1 when double-buffer is used. 0: If either timer register 0 or timer register 1 is prepared to be written, one register can be written at a time. 1: If both timer register 0 and timer register 1 are not prepared, timer register cannot be written.
5	MTTBCP	W	Controls software capture 0: Capture count values to the capture register 0 (MTxCP0) 1: Don't care
4-3	MTTBCPM[1:0]	R/W	Sets capture timing 00: Capture is disabled. 01: At the rising edge of MTTBxIN input, counter values are captured to the capture register 0 (MTxCP0) 10: At the rising edge of MTTBxIN input, counter values are captured to the capture register 0 (MTxCP0). At the falling edge of MTTBxIN input, counter values are captured to the capture register 1 (MTxCP1). 11: Capture is disabled.
2	MTTBCLE	R/W	Clear MPT up-counter 0: Clear is disabled. 1: Clear MPT up-counter by matching with timer register 1 (MTxRG1)
1-0	MTTBCLK[1:0]	R/W	Selects timer count clock of MPT 00: MTTBxIN input 01: φT1 10: φT4 11: φT16

Note 1: **MTxTBMOD<MTTBCP> reads as "1".**

Note 2: **Do not modify MTxTBMOD during timer in operation (MTxRUN<MTRUN> = "1").**

10.4.7 MTxTBFFCR (MPT flip-flop control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	MTTBC1T1	MTTBC0T1	MTTBE1T1	MTTBE0T1	MTTBFF0C	
After reset	1	1	0	0	0	0	1	1

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-6	-	R	Read as "11".
5	MTTBC1T1	R/W	Controls timer flip-flop reverse when up-counter values are captured to the capture register 1 (MTxCP1). 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
4	MTTBC0T1	R/W	Controls timer flip-flop reverse when up-counter values are captured to the capture register 0 (MTxCP0) 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
3	MTTBE1T1	R/W	Controls timer flip-flop reverse when up-counter values and the timer register 1 (MTxRG1) are matched. 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
2	MTTBE0T1	R/W	Controls timer flip-flop reverse when up-counter values and the timer register 0 (MTxRG0) are matched. 0: MTxFF0 does not reverse. 1: MTxFF0 reverses.
1-0	MTTBFF0C	R/W	Controls timer flip-flop 00: Reverses a value of MTxFF0. 01: Sets "1" to MTxFF0. 10: Sets "0" MTxFF0 to clear. 11: Don't care. Read as "11".

Note: Do not modify **MTxTBFFCR during timer in operation (MTxRUN<MTRUN> = "1").**

10.4.8 MTxTBST (MPT status register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	MTTBINT TBOF	MTTBINTTB1	MTTBINTTB0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTTBINTTBOF	R	Indicates the status of up-counter overflow interrupt generation. 0: No interrupt generation 1: Interrupt generation
1	MTTBINTTB1	R	Indicates the interrupt generation by matching with timer register 1 (MTxRG1) 0: No interrupt generation 1: Interrupt generation
0	MTTBINTTB0	R	Indicates the interrupt generation by matching with timer register 0 (MTxRG0) 0: No interrupt generation 1: Interrupt generation

Note: **Once any interrupt generates, corresponding flag in MTxTBST register is set to notify CPU of an interrupt generation. If MTxTBST register is read, the flag is cleared to "0".**

10.4.9 MTxTBIM (MPT interrupt mask register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-					MTTBIMOF	MTTBIM1	MTTBIM0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTTBIMOF	R/W	Controls up-counter overflow interrupt to mask 0: Not mask interrupt 1: Masks interrupt
1	MTTBIM1	R/W	Controls to mask the interrupt when the match between timer register 1 (MTxRG1) and up-counter. 0: Not mask interrupt 1: Masks interrupt
0	MTTBIM0	R/W	Controls to mask the interrupt when the match between timer register 0 (MTxRG0) and up-counter. 0: Not mask interrupt 1: Masks interrupt

Note: MTxTBST reflects interrupt requests even though **MTxTBIM masks interrupts.**

10.4.10 MTxTBUC (MPT read capture register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTUC[15:0]	R	Reading MTxTBUC captures the current up-counter value.

10.4.11 MTxRG0/MTxRG1 (MPT timer register)

MTxRG0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG0[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG0[15:0], match detection interrupt (INTMTTBx0) occurs. Also, MTTBxOUT can be reversed when matching, [IGBT mode] When up-counter values match with MTRG0[15:0], MTOUT0x becomes active level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0<MTxRG0<MTxRG1≤MTxIRG4≤0xFFFF.**

MTxRG1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG1[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG1[15:0], match detection interrupt (INTMTTBx1) occurs. Also, MTTBxOUT can be reversed when matching. [IGBT mode] When up-counter values match with MTRG1[15:0], MTOUT0x becomes inactive level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0<MTxRG0<MTxRG1≤MTxIRG4≤0xFFFF.**

10.4.12 MTxCP0 /MTxCP1 (MPT capture register)

MTnCP0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTCP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP0[15:0]	R	Read captured up-counter values.

MTnCP1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTCP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP1[15:0]	R	Read captured up-counter values.

Note: **During the timer stopping, a value of timer counter (MTUCx) cannot be read. When the timer stops, a value previously captured is held and the value can be read.**

10.5 Operational Description categorized by circuit

10.5.1 Prescaler

This 4-bit prescaler generates the source clock for up-counter MTUCx.

Input clock φT0 to the prescaler is chosen among fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 and fperiph/32 by specifying with CGSYSCR<PRCK[2:0]>. This peripheral clock (fperiph) is either fgear specified with CG SYSCR<FPSEL[1:0]> or fc that is pre-divided clock gear.

Prescaler is set to enable/disable with MTxRUN<MTPRUN>. When MTxRUN<MTPRUN> is set to "1", counting starts. When MTxRUN<MTPRUN> is set to "0", the counter is stopped and cleared. Table 10-2 shows prescaler output clock resolutions.

Table 10-2 Prescaler output clock resolutions (fc = 40MHz)

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

Table 10-2 Prescaler output clock resolutions (fc = 40MHz)

Peripheral clock selection <FPSEL[1:0]>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function		
			φT1	φT4	φT16
01 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^{1}$ (0.05 μs)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)
		001 (fperiph/2)	$fc/2^{2}$ (0.1 μs)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		101 (fperiph/32)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	100 (fc/2)	000 (fperiph/1)	–	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)
		001 (fperiph/2)	$fc/2^{2}$ (0.1 μs)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		101 (fperiph/32)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	–	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)
		001 (fperiph/2)	–	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		101 (fperiph/32)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	$fc/2^{5}$ (0.8 μs)
		001 (fperiph/2)	–	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)
		010 (fperiph/4)	–	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		101 (fperiph/32)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	$fc/2^{5}$ (0.4 μs)
		001 (fperiph/2)	–	–	$fc/2^{6}$ (0.8 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: **Prescaler output clock φTn must satisfy the condition of φTn <fsys. (φTn must be slower than fsys.)**

Note 2: **Do not change the clock gear during timer in operation.**

Note 3: **In the above table, "−" indicates prohibition.**

10.5.2 Up-Counter (MTUC0)

This counter is a 16-bit binary counter.

- Source clock

  The source clock can be set with MTxTBMOD<MTTBCLK[1:0]>.

  Prescaler output clock can choose among φT1, φT4, φT16 or external clock of MTTBxIN pin.

- Start/Stop counter operation

  Counter operation is set with MTxRUN<MTRUN>. When <MTRUN> = "1" is set, counter operation starts. When <MTRUN> = "0" is set, the counter is stopped and cleared at the same time.

  When a value of up-counter MTUCx detects the match with a setting value of timer register MTxRG0/MTxRG1, INTMTTB0x or INTMTTB1x occurs.

- Counter clear timing

  1. Comparing a match

     If MTxTBMOD<MTTBCLE> = "1" is set, the counter is cleared when comparing matches with MTxRG1.

     If MTxTBMOD<MTTBCLE> = "0" is set, the counter becomes a free-running counter.

  2. Counter stopping

     If MTxRUN<MTRUN> = "0" is set, the counter is stopped and cleared.

- Overflow of the counter

  If MTUCx is overflowed, an overflow interrupt INTMTTB0x occurs.

10.5.3 Timer Register (MTxRG0, MTxRG1)

Timer register sets a values to compare with up-counter MTUCx. Comparator compares a value of timer register with a value of up-counter. If these two are matched, the match detection signal is output.

- Structure

  In the timer register, MTxRG0/1 is double-buffering structure paired with register buffer.

  Double-buffer is set to enable/disable with MTxTBCR<MTTBWBF>. If <MTTBWBF> = "0" is set, double-buffer is disabled. If <MTTBWBF> = "1" is set, double-buffer is enabled.

  While double-buffer is enabled, data transfer is taken place from register buffer 0 to timer register MTxRG0/1 when MTUCx matches with MTxRG1.

- Initial state

  After reset, MTxRG0 and MTxRG1 are undefined and double-buffer is disabled.

- How to set

  1. If double-buffer is not used.

     Use half-word access or word access

  2. If double-buffer is used.

     MTxRG0 and 1, and register buffer 0 and 1 are assigned to the same address respectively.

     When <MTTBWBF> is "0", MTxRG0 and 1 and each register buffer are written the same value. When <MTTBWBF> is "1", only corresponding register buffer is written data. Thus when writing the initial value to timer register, set as follows; firstly register buffer is disabled, secondly timer register is written data, thirdly <MTTBWBF> is set to "1". Finally next data is written to register buffer.

10.5.4 Capture Control

This circuit controls the timing when a value of up-counter MTUCx is latched by capture register MTxCP0/MTxCP1. This latch timing is set with MTxTBMOD<MTTBCPM[1:0]>.

Also the timing is controlled by software. Every time MTxTBMOD<MTTBCP> is set to "0", a value of MTUCx is captured to the capture register MTxCP0 at the time. Note that prescaler must be set to RUN status (MTxRUN<MTPRUN> = "1").

10.5.5 Capture Register (MTxCAP0、MTxCAP1)

This register captures a value of up-counter MTUCx.

10.5.6 Up-counter Capture Register (MTxTBUC)

Besides the capture function using capture control circuit, current counter value of up-counter (MTUC0) is also captured by reading MTxTBUC register.

10.5.7 Comparators (CP0, CP1)

This comparator detects the match comparing a value of up-counter (MTUCx) with a setting value of timer register MTxRG0/MTxRG1. If these values are matched, INTMTTBx0 or INTMTTBx1 occurs.

10.5.8 Timer Flip-flop (MTxFF0)

Timer flip-flop circuit (MTxFF0) reverses by a match signal from comparators or a latch signal to the capture register. This reverse is enabled/disabled with MTxTBFFCR<MTTBC1T1, MTTBC0T1, MTTBE1T1, MTTBE0T1>.

After reset, a value of MTxFF0 is undefined. If MTxTBFFCR<MTTBFF0C[1:0]> is set to "00", the reverse is enabled. If MTxTBFFCR<MTTBFF0C[1:0]> is set to "01", MTxFF0 is set to "1". MTxTBFFCR<MTTBFF0C[1:0]> is set to "10", MTxFF0 is set to "0" to clear.

A value of MTxFF0 can be output to timer output pin MTTBxOUT. If timer output is used, port related registers (PxCR and PxFR) must be set beforehand.

10.5.9 Capture Interrupts (INTMTCAPx0, INTMTCAPx1)

Capture interrupts (INTMTCAPx0 and INTMTCAPx1) occur respectively at the timing when data is latched to each capture register (MTxCP0 and MTxCP1). Interrupt setting is set by CPU.

10.6 Operational Description in IGBT mode

10.6.1 Block Diagram

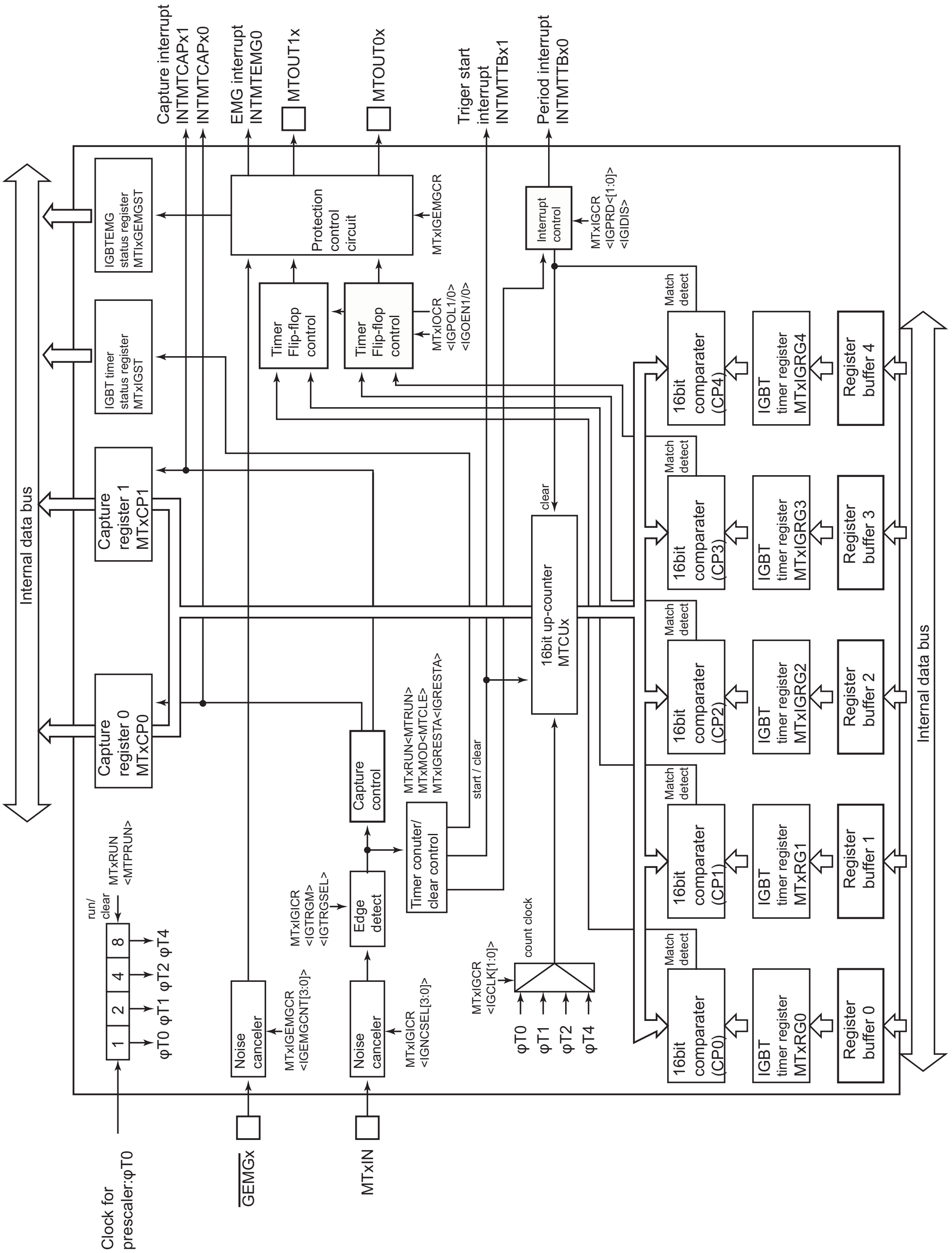


Figure 10-3 Block Diagram in IGBT mode

10.6.2 Registers in IGBT mode categorized by channel

This section describes registers and addresses of each channel.

Channel x	Base Address
MPT 0	0x4005_0800
MPT 1	0x4005_0880
MPT 2	0x4005_0900

Register name(x = 0 to 2)		Address (Base+)
MPT enable register	MTxEN	0x0000
MPT RUN register	MTxRUN	0x0004
MPT register	MTxRG0 MTxRG1	0x0020 0x0024
MPT capture register	MTxCP0 MTxCP1	0x0028 0x002C
IGBT control register	MTxIGCR	0x0030
IGBT timer restart register	MTxIGRESTA	0x0034
IGBT timer status register	MTxIGST	0x0038
IGBT input control register	MTxIGICR	0x003C
IGBT output control register	MTxIGOCR	0x0040
IGBT timer register 2, 3, 4	MTxIGRG2 MTxIGRG3 MTxIGRG4	0x0044 0x0048 0x004C
IGBT EMG control register	MTxIGEMGCR	0x0050
IGBT EMG status register	MTxIGEMGST	0x0054

10.6.3 MTxEN (MPT enable register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	MTEN	R/W	Specifies MPT operation. 0: Disabled 1: Enabled When MTEN is disabled, feeding clock to other registers of MPT module is stopped, so that power consumption can be reduced. (Read or write to other registers cannot be done.)
6	MTHALT	R/W	Specifies MPT operation when core halts (debug break). [TMR function] 0: Clock stopping operation is disabled while core halts. 1: Clock stopping operation is enabled while core halts. [IGBT function] 0: Not control clock stopping operation and MTOUT0x/MTOUT1x output. 1: Clock stopping operation is enabled while core halt. It controls MTOUT0x/MTOUT1x output according to the MTxIGEMGCR<IGEMGOC> setting.
5-1	-	R	Read as "0".
0	MTMODE	R/W	Specifies operation mode. 0: Timer mode 1: IGBT mode

Note: When MPT is used, MPT operation is enabled (<MTEN> = "1")before each register of MPT module is set. If MPT operation is disabled after MPT is stopped, each register setting is maintained.

10.6.4 MTxRUN (MPT RUN register)

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

Bit	Bit Symbol	Type	Function
31-3	-	R	Read as "0".
2	MTPRUN	R/W	Controls MPT prescaler operation 0: Stops prescaler operation. Prescaler is cleared to "0". 1: Starts prescaler operation.
1	-	R	Read as "0".
0	MTRUN	R/W	Controls MPT counting operation 0: Stops counting operation. Counter is cleared to "0". 1: Starts counting operation.

10.6.5 MTxRG0/MTxRG1 (MPT timer register)

MTxRG0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTRG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG0[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG0[15:0], match detection interrupt (INTMTTBx0) occurs. Also, when matching, MTTBxOUT can be reversed. [IGBT mode] When up-counter values match with MTRG0[15:0], MTOUT0x becomes active level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0<MTxRG0<MTxRG1≤MTxIRG4≤0xFFFF.**

MTxRG1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTRG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTRG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTRG1[15:0]	R/W	Timer count value [Timer mode] When up-counter values match with MTRG1[15:0], match detection interrupt (INTMTTBx1) occurs. Also, MTTBxOUT can be reversed when matching. [IGBT mode] When up-counter values match with MTRG1[15:0], MTOUT0x becomes inactive level.

Note 1: **Use half word access or word access.**

Note 2: **Set to the condition of 0<MTxRG0<MTxRG1≤MTxIRG4≤0xFFFF.**

10.6.6 MTxCP0 /MTxCP1 (MPT capture register)

MTnCP0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTCP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP0[15:0]	R	Read captured up-counter values.

MTnCP1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MTCP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MTCP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MTCP1[15:0]	R	Read captured up-counter values.

Note: **During the timer stopping, a value of timer counter (MTUCx) cannot be read. When the timer stops, a value previously captured is held and the value can be read.**

10.6.7 MTxIGCR (IGBT control register)

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

Bit	Bit Symbol	Type	Function
31-11	-	R	Read as "0".
10	IGIDIS	R/W	Controls an interrupt when commands start 0: Enabled 1: Disabled
9-8	IGPRD[1:0]	R/W	Chooses an interrupt cycle 00: Every one cycle 01: Every two cycles 10: Every four cycles 11: Reserved
7	-	R	Read as "0".
6	IGSNGL	R/W	Chooses IGBT operation 0: Continuous operation 1: Single operation
5-4	IGSTP[1:0]	R/W	Chooses stopping status 00: Initial output status and counter immediately stops to clear 01: Sustains output status and counter immediately stops to clear 10: After cycle time has elapsed then counter stops to clear 11: Reserved
3-2	IGSTA[1:0]	R/W	Chooses start mode 00: Command start and trigger capture 01: Command start and trigger start 10: Trigger start 11: Reserved
1-0	IGCLK[1:0]	R/W	Chooses a source clock of IGBT 00: φT0 01: φT1 10: φT2 11: φT4

Note 1: **Do not modify MTxIGCR during timer in operation (MTxRUN<MTRUN> = "1").**

Note 2: **When the counter stops after specified cycle time has elapsed, or counter is stopped with (MTxIGCR <IGSTP> = "10") and cleared with MTxRUN<MTRUN>, check if the timer is stopped by cycle interrupt generation. Then change the setting and restart.**

10.6.8 MTxIGRESTA (IGBT timer restart register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	IGRESTA	W	Controls counting restart 0: Don't care 1: Restart Read as "0".

Note: **If MTxIGRESTA<IGRESTA> is set to "1" during timer in operation, timer counter can be cleared and restart. Please check the status of output waveform before setting is changed.**

10.6.9 MTxIGST (IGBT timer status register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	IGST	R	Counter operation status 0: Stop 1: Operating

10.6.10 MTxIGICR (IGBT input control register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	IGTRGM	R/W	Controls trigger edge accept mode 0: Always accept 1: Acceptance is disabled during active level
6	IGTRGSEL	R/W	Chooses start trigger edges and its active levels. 0: Rising edge start and active level is "High". 1: Falling edge start and active level is "Low".
5-4	-	R	Read as "0".
3-0	IGNCSEL[3:0]	R/W	Trigger input noise elimination time selection Noise elimination time is calculated with the following formula: IGNCSEL[3:0] × 16 / fsys 0000: Noise filter is not used. 0001: Noise elimination time 16 / fsys [s] 0010: Noise elimination time 32 / fsys [s] 0011: Noise elimination time 48 / fsys [s] 0100: Noise elimination time 64 / fsys [s] 0101: Noise elimination time 80 / fsys [s] 0110: Noise elimination time 96 / fsys [s] 0111: Noise elimination time 112 / fsys [s] 1000: Noise elimination time 128 / fsys [s] 1001: Noise elimination time 144 / fsys [s] 1010: Noise elimination time 160 / fsys [s] 1011: Noise elimination time 176 / fsys [s] 1100: Noise elimination time 192 / fsys [s] 1101: Noise elimination time 208 / fsys [s] 1110: Noise elimination time 224 / fsys [s] 1111: Noise elimination time 240 / fsys [s]

Note 1: **Do not modify MTxIGCR during timer in operation (MTxRUN<MTRUN> = "1").**

Note 2: **When MTxGCR<IGNCSEL[3:0]> is used, EMG protection circuit must be disabled (MTxIGEMGCR<IGEMGEN> = "0").**

Note 3: **When MTxIGCR<IGNCSEL[3:0]> is changed, specified noise elimination time or more is required to start the timer with (MTxRUN<MTRUN> = "1").**

10.6.11 MTxIGOCR (IGBT output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	IGPOL1	IGPOL0	-	-	IGOEN1	IGOEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-6	-	R	Read as "0".
5	IGPOL1	R/W	Indicates the initial state of MTOUT1x 0: Low 1: High
4	IGPOL0	R/W	Indicates the initial state of MTOUT0x 0: Low 1: High
3-2	-	R	Read as "0".
1	IGOEN1	R/W	Controls MTOUT1x output 0: Disable 1: Enable
0	IGOEN0	R/W	Control MTOUT0x output 0: Disabled 1: Enabled

Note: **MTOUT0x/MTOUT1x output is changing according to a content of IGBT output control register (MTxIGOCR) regardless of timer in operation/stopping. Check the operation status before MTxGOCR is set.**

10.6.12 MTxIGRG2 (IGBT timer register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	IGRG2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IGRG2							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	IGRG2[15:0]	R/W	Timer count value When up-counter matches with IGRG2[15:0], MTOUT1x becomes active level.

Note 1: **Use half-word access or word access.**

Note 2: **Set the value to the condition of 0<MTxIGRG2<MTxIGRG3≤MTxIGRG4≤0xFFFF.**

10.6.13 MTxIGRG3 (IGBT timer register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	IGRG3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IGRG3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	IGRG3[15:0]	R/W	Timer count value When up-counter matches with IGRG3[15:0], MTOUT1x becomes inactive level.

Note 1: **Use half-word access or word access.**

Note 2: **Set the value to the condition of 0<MTxIGRG2<MTxIGRG3≤MTxIGRG4≤0xFFFF.**

10.6.14 MTxIGRG4 (IGBT timer register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	IGRG4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IGRG4							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	IGRG4[15:0]	R/W	Timer count value Specifies IGBT mode cycle

Note 1: **Use half-word access or word access.**

Note 2: **Set the value to the condition of 0<MTxRG0<MTxRG1≤MTxIGRG4≤0xFFFF.**

Note 3: **Set the value to the condition of 0<MTxIGRG2<MTxIGRG3≤MTxIGRG4≤ 0xFFFF.**

10.6.15 MTxIGEMGCR (IGBT EMG control register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	IGEMGCNT[3:0]	R/W	GEMG input noise elimination time selection Noise elimination time is calculated with the following formula: IGEMGCNT[3:0]×16 / fsys 0000: Noise filter is not used. 0001: Input noise elimination time 16 / fsys [s] 0010: Input noise elimination time 32 / fsys [s] 0011: Input noise elimination time 48 / fsys [s] 0100: Input noise elimination time 64 / fsys [s] 0101: Input noise elimination time 80 / fsys [s] 0110: Input noise elimination time 96 / fsys [s] 0111: Input noise elimination time 112 / fsys [s] 1000: Input noise elimination time 128 / fsys [s] 1001: Input noise elimination time 144 / fsys [s] 1010: Input noise elimination time 160 / fsys [s] 1011: Input noise elimination time 176 / fsys [s] 1100: Input noise elimination time 192 / fsys [s] 1101: Input noise elimination time 208 / fsys [s] 1110: Input noise elimination time 224 / fsys [s] 1111: Input noise elimination time 240 / fsys [s]
3	-	R	Read as "0".
2	IGEMGRS	W	Return from EMG protection status 0: Don't care 1: Returned (automatically cleared to "0".) (Read as "0".)
1	IGEMGOC	R/W	Set the polarity of MTOUT0x/MTOUT1x at EMG protection 0: Inactive level 1: High-impedance
0	IGEMGEN	R/W	Controls EMG protection circuit operation 0: Disable 1: Enable

Note: **Do not modify MTxIGEMGCR during timer in operation (MTxRUN<MTRUN> = "1").**

10.6.16 MTxIGEMGST (IGBT EMG status register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	IGEMGIN	R	EMG input status after noise elimination 0: Low 1: High
0	IGEMGST	R	EMG protection status 0: Normal operation 1: During in protection Read value indicates EMG protection status

10.7 Operation Description categorized by circuit

10.7.1 Prescaler

This 4-bit prescaler generates the source clock for up-counter MTUCx.

Input clock φT0 to the prescaler is chosen among fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 and fperiph/32 by specifying with CGSYSCR<PRCK[2:0]>. This peripheral clock (fperiph) is either fgear specified with CGSYSCR<FPSEL[1:0]> or fc that is a pre-dividing clock gear.

Prescaler is set to enable/disable with MTxRUN<MTPRUN>. When MTxRUN<MTPRUN> is set to "1", counting starts. When MTxRUN<MTPRUN> is set to "0", the counter is stopped and cleared. Table 10-3 shows prescaler output clock resolutions.

Table 10-3 Prescaler output clock resolutions (fc = 40MHz)

Peripheral clock selection <FPSEL[1:0]>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function			
			φT0	φT1	φT2	φT4
00 (fgear)	000 (fc)	000 (fperiph/1)	fc (0.025 μs)	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)
		001 (fperiph/2)	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)
		010 (fperiph/4)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		011 (fperiph/8)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		100 (fperiph/16)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		101 (fperiph/32)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)
		001 (fperiph/2)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		010 (fperiph/4)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		011 (fperiph/8)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		001 (fperiph/2)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		010 (fperiph/4)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		011 (fperiph/8)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
		100 (fperiph/16)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)
		101 (fperiph/32)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)	fc/2^{10} (25.6 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		001 (fperiph/2)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		010 (fperiph/4)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
		011 (fperiph/8)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)
		100 (fperiph/16)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)	fc/2^{10} (25.6 μs)
		101 (fperiph/32)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{8} (6.4 μs)	fc/2^{9} (12.8 μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{9} (12.8 μs)	fc/2^{10} (25.6 μs)	fc/2^{11} (51.2 μs)	fc/2^{12} (102.4 μs)

Table 10-3 Prescaler output clock resolutions (fc = 40MHz)

Peripheral clock selection <FPSEL[1:0]>	Clock gear value <GEAR[2:0]>	Prescaler clock selection <PRCK[2:0]>	Prescaler output clock function			
			φT0	φT1	φT2	φT4
01 (fc)	000 (fc)	000 (fperiph/1)	fc (0.025 μs)	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)
		001 (fperiph/2)	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)
		010 (fperiph/4)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		011 (fperiph/8)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		100 (fperiph/16)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		101 (fperiph/32)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)
		001 (fperiph/2)	fc/2^{1} (0.05 μs)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)
		010 (fperiph/4)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		011 (fperiph/8)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		100 (fperiph/16)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		101 (fperiph/32)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
	101 (fc/4)	000 (fperiph/1)	–	–	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)
		001 (fperiph/2)	–	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)
		010 (fperiph/4)	fc/2^{2} (0.1 μs)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		011 (fperiph/8)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		100 (fperiph/16)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		101 (fperiph/32)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	–	fc/2^{3} (0.2 μs)
		001 (fperiph/2)	–	–	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)
		010 (fperiph/4)	–	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		011 (fperiph/8)	fc/2^{3} (0.2 μs)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		100 (fperiph/16)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		101 (fperiph/32)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	–	–
		001 (fperiph/2)	–	–	–	fc/2^{4} (0.4 μs)
		010 (fperiph/4)	–	–	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)
		011 (fperiph/8)	–	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)
		100 (fperiph/16)	fc/2^{4} (0.4 μs)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)
		101 (fperiph/32)	fc/2^{5} (0.8 μs)	fc/2^{6} (1.6 μs)	fc/2^{7} (3.2 μs)	fc/2^{8} (6.4 μs)

Note 1: **Prescaler output clock φTn must satisfy the condition of φTn < fsys. (φTn must be slower than fsys.)**

Note 2: **Do not change the clock gear during timer in operation.**

Note 3: **In the above table, "−" indicates prohibition.**

10.7.2 Up-Counter (MTUCx)

This counter is a 16-bit binary counter.

- Source clock

  The source clock can be set with MTxIGCR<IGCLK[1:0]>.

  Prescaler output clock can be chosen among φT0, φT1, φT2, φT4.

- Start/Stop counter operation

  Counter operation is set with MTxRUN<MTRUN>. When <MTRUN> = "1" is set, counter operation starts. When <MTRUN> = "0" is set, the counter is stopped and cleared at the same time.

  And when MTxIGRESTA<IGRESTA> = "1" is set, counter is cleared and started count-up from zero.

- Counter clear timing

  1. Comparing a match

     The counter is cleared when a value of up-counter (MTUCx) is match with MTxIGRG4.

  2. Counter stopping

     If Mx0RUN<MTRUN> = "0" is set, the counter is stopped and cleared.

  3. Counter restarts

     If MTxIGRESTA<IGRESTA> = "1" is set, the counter is cleared and counted-up from 0.

  4. In trigger start mode

     In trigger start mode, the counter is stopped and cleared when MTxIN pin becomes the stopping to clear level.

- Count-up & clear operation

  Count-up & clear operation and setting cycle are described in the two cases respectively; one is the case that φT0 is chosen as a source clock, the other case is that φT1, φT2 or φT4 is chosen as a source clock.

  1. φT0 is selected as a source clock

     When φT0 is selected as a source clock, two source clocks are required for match counting and clear counting, so that setting cycle is M+1.

  2. φT1, φT2 or φT4 is selected as a source clock

     When φT1, φT2 or φT4 is selected as a source clock, one source clock is required for match counting and clear counting, so that setting cycle is M.

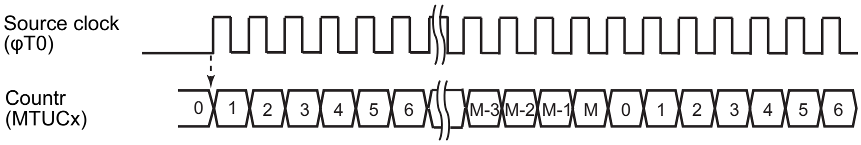


Figure 10-4 Count-up/clear operation when φT0 is selected as a source clock

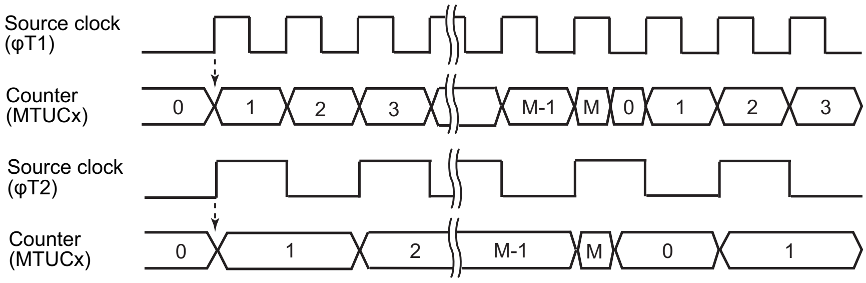


Figure 10-5 Count-up/clear operation when φT1, φT2 or φT4 is selected as a source clock

10.7.3 Cycle Setting Register (MTxIGRG4)

This register sets the cycle of PPG output consisting of double-buffering structure. Data update timing is one cycle after when MTxIGRG4 matches with up-counter MTUCx clearing the counter. At this time, data transfer is taken place from register buffer 4 to timer register MTxIGRG4.

10.7.4 Timer register (MTxRG0, MTxRG1, MTxIGRG2, MTxIGRG3, MTxIGRG4)

This register sets a value to compare with up-counter MTUCx. When these are matched, the match detect signal is output. Timer register, MTxRG0/1, MTxIGRG2/3 are double-buffering structure paired with each register buffer. When MTxIGRG4 matches with up-counter MTUCx, the counter is cleared and data is updated at the same time. Also at this time, data transfer is taken place from register buffer 2/3 to timer register MTxIGRG2/3.

In IGBT mode, MTxRG0/1 is always double-buffering structure.

- Write/read operation of timer registers (MTxRG0, MTxRG1, MTxIGRG2 and MTxIGRG3) and cycle register (MTxIGRG4)

  1. Write

     When timer is stopping, above registers can be written directly. In timer in operation, data is latched in each register. When MTxIGRG4 matches with up-counter MTUCx, the counter is cleared and data is updated at the same time.

  2. Read

     Read the current value of target register comparing with 16-bit comparator. A value of register buffer cannot be read.

Note: **Use half-word access or word access.**

10.7.5 Capture Control

If command start or capture mode is set, this circuit captures up-counter values (MTUCx) at the rising and falling edges of MTxIN to MTxCP0 and MTxCP1 respectively.

10.7.6 Capture Register (MTxCAP0, MTxCAP1)

This register captures a value of up-counter MTUCx.

10.7.7 Comparators (CP0, CP1, CP2, CP3, CP4)

This comparator detects the match comparing a value of up-counter (MTUCx) with a setting value of timer register MTxRG0, MTxRG1, MTxIGRG2, MTxIGRG3 and MTxIGRG4.

10.7.8 MTOUT0x, MTOUT1x Output Control

When up-counter matches with timer register, MTOUT0x or MTOUT1x is output.

Initial setting of output pin is set with MTxIGOCR<IGPOL[1:0]>. After reset, initial state is low. When MTxIGOCR<IGPOL[1:0]> = "00" is set, initial state is low. When MTxIGOCR<IGPOL[1:0]> = "11" is set, initial state is high. Output control is set with MTxIGOCR<IGOEN0[1:0]>. After reset, MTxIGOCR<IGOEN [1:0]> is disabled. If MTxIGOCR<IGOEN[1:0]> is enabled, set to "1".

10.7.9 Capture Interrupts (INTMTCAPx0,INTMTCAPx1)

Capture interrupts (INTMTCAPx0 and INTMTCAPx1) occur respectively when each capture register (MTxCP0 and MTxCP1) latches data. Interrupt setting is set by CPU.

10.7.10 Trigger Start Interrupt (INTMTTBx1)

When command start & trigger start mode or trigger start mode is chosen, trigger interrupt occurs when the edge specified with MTxIGCR<IGTRGSEL> is input and the counter starts. In the trigger capture mode, INTMTTBx1 interrupt does not generate at the trigger edge. When emergency output is stopping, a start trigger interrupt occurs.

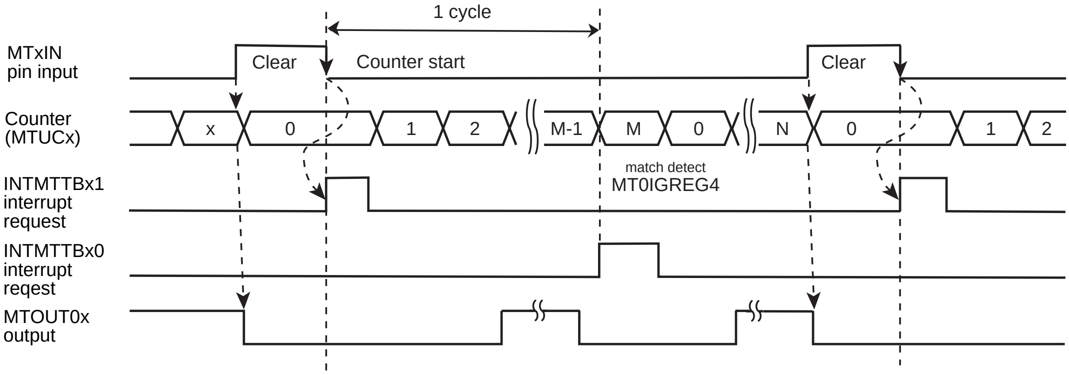


Figure 10-6 Trigger start interrupt operation

10.7.11 Cycle Interrupt (INTMTTBx0)

When command start & trigger capture mode or command start & trigger start mode is chosen, a cycle interrupt occurs when count starts in command start or counter reaches to a value of counter cycle setting (MTxIGREG4) (cycle finishes by matching with a value of cycle setting). Also, a cycle interrupt occurs by matching with a value of counter cycle when emergency output is stopping. Interrupt cycle can be set to among every one cycle, every two cycles or every four cycles with MTxIGCR<IGPRD[1:0]>.

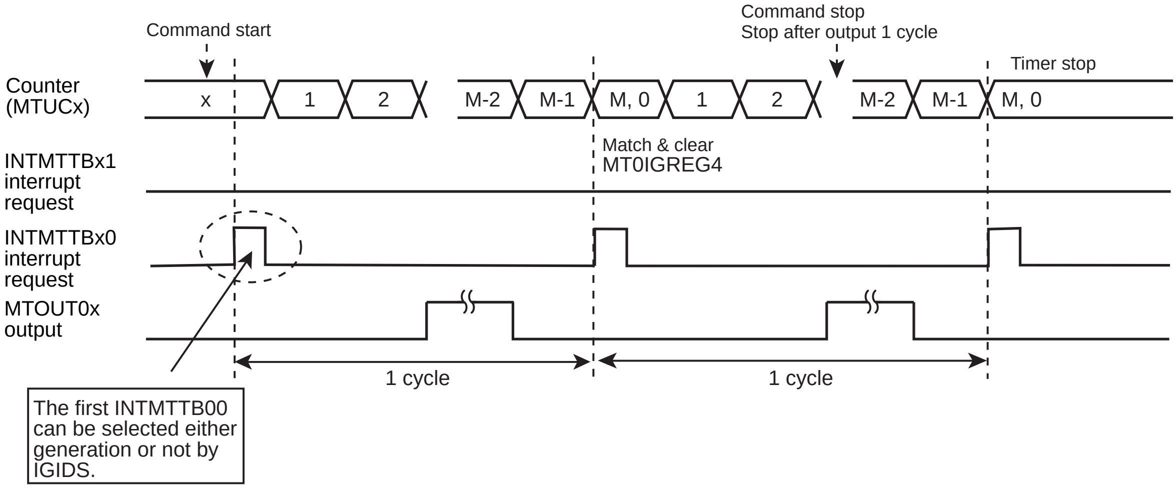


Figure 10-7 Cycle interrupt operation

In command start, a cycle interrupt at the starting count is set to enable/disable with interrupt control register MTxIGCR<IGIDIS>. At starting command (MTxRUN<MTRUN> is set to "1"), if MTxIN pin is stopping level, counting does not start (INTMTTBx0 does not occur). Counting starts by trigger start edge and NTMTTBx1 occurs.

10.7.12 Basic Operation

Each MTOUT0x pin and MTOUT1x pin output PPG.

This circuit controls waveform by comparing data set in the timer register (MTxRG0/1, MTxIGRG2/3/4) with a value of 16-bit up-counter.

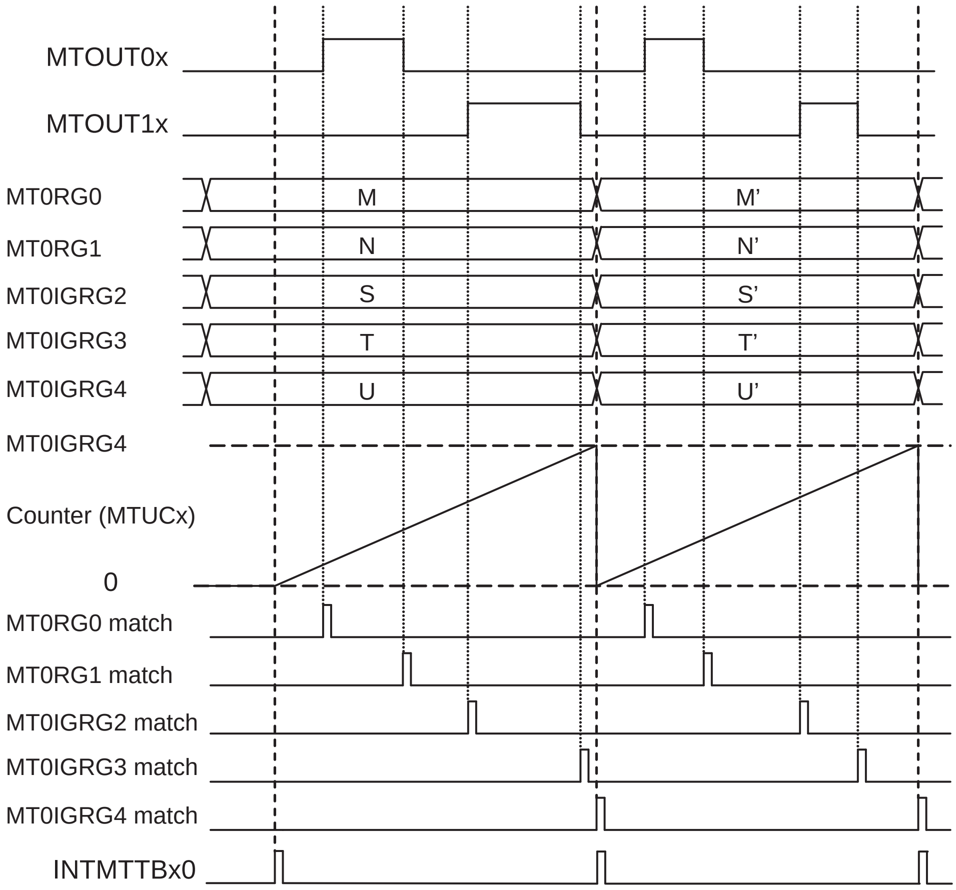


Figure 10-8 IGBT mode basic timing

10.7.13 Start modes

In IGBT mode, three start modes are available.

10.7.13.1 Command Start & Trigger Capture Mode

When MTxRUN<MTRUN> is set to "1", counting-up starts. If the counter reaches to the setting cycle, the counter is cleared. At this time, continuous mode is set with MTxIGCR<IGSNGL>, count-up starts again. If single mode is set, counting stops.

If MTxIGRESTA<IGRESTA> is set to "1" before reaching to the setting cycle, counter is cleared at this point and count-up continues.

Counter value at the rising edge/falling edge of MTxIN input can be stored to capture registers.

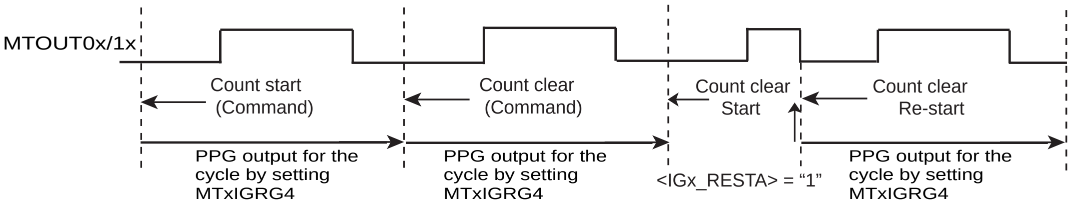


Figure 10-9 Continuous mode in command start

In the command start & trigger capture mode, when the counter starts, a counter value is captured at the each rising/falling edge of MTxIN input to each capture register (MTxCAP0 and MTxCAP1).When capture operation is done, INTMTCAPx0 and INTMTCAPx1 occur at each edge.

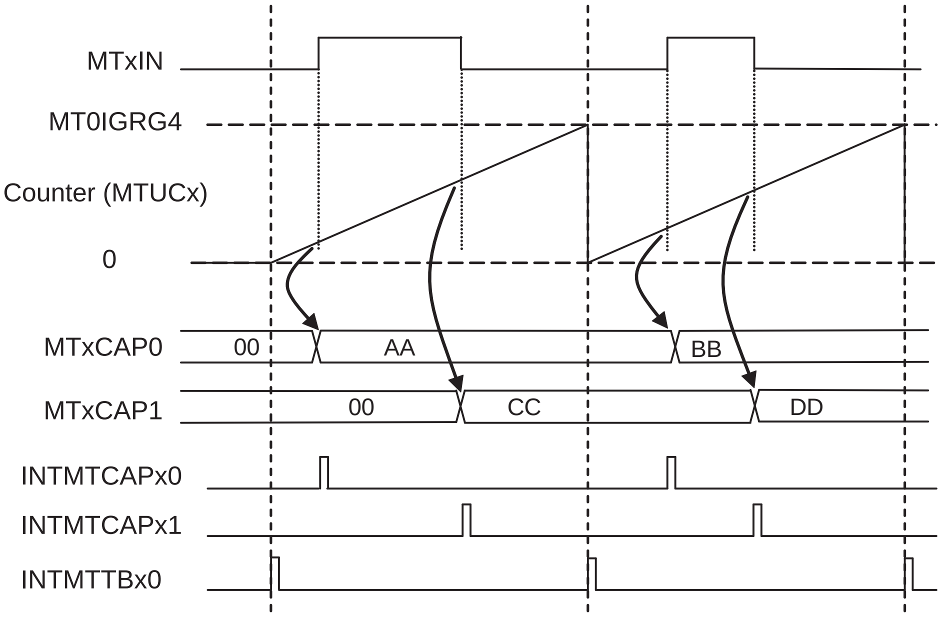


Figure 10-10 Capture operation

10.7.13.2 Command Start & Trigger Start Mode

When MTxRUN<MTRUN> is set to "1", count-up starts. If there is no trigger inputs to MTxIN input, same operation previously described in command start & capture mode starts. If an edge input specified with MTxIGICR<IGTRGSEL> to MTxIN pin exists, timer counting starts. While specified clear stopping level is input, the counter is not cleared. At the starting command (when MTxRUN<MTRUN> is set to "1"), if MTxIN pin is in the stopping level, the counter does not start (INTMTTBx1 does not generate). Counting starts by trigger start edge and INTMTTBx1 occurs. (Trigger input is prior to command start.)

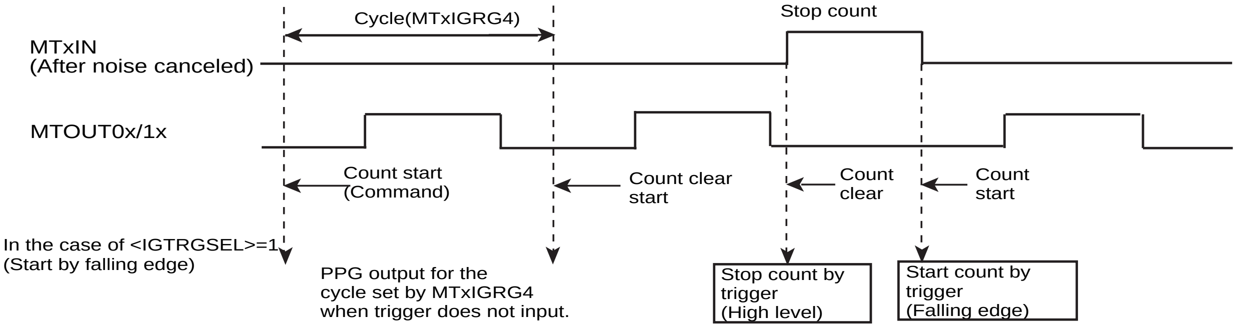


Figure 10-11 Command start & trigger start

10.7.13.3 Trigger Start Mode

If an edge input specified with MTxIGICR<IGTRGSEL> exists, timer counting starts. While specified clear stopping level is input, the counter is not cleared.

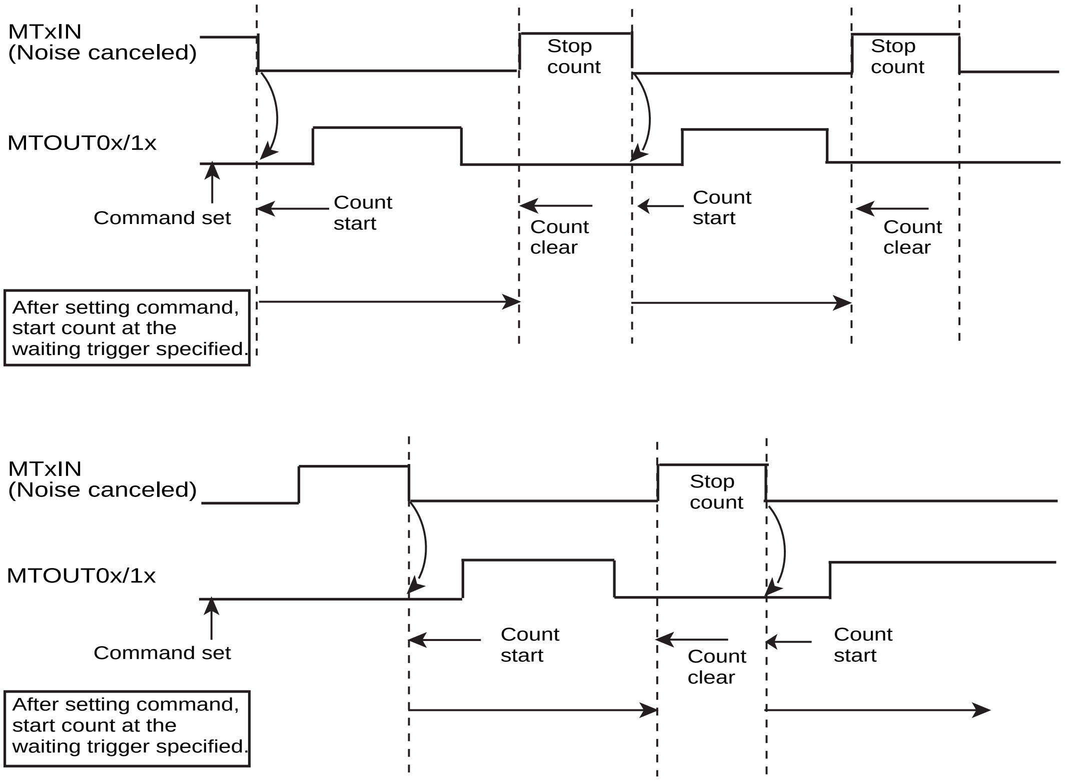


Figure 10-12 Trigger Start

10.7.14 Single/Continuous Output Mode

Single/continuous output mode can be set with IGBT output.

10.7.14.1 Continuous Output Mode

At the starting timer (MTxRUN<MTRUN> = "1"), if MTxIGCR<IGSNGL> = "0" is set, continuous output mode is chosen. In the continuous output mode, specified continuous waveform can be output.

10.7.14.2 Single Output Mode

At the starting timer (MTxRUN<MTRUN> = "1"), if MTxIGCR<IGSNGL> = "1" is set, single output mode is chosen. In the single output mode, the counter stops after output every single cycle.

At the trigger starting, the counter stops until triggers are input. Counting starts by the specified trigger input, and after one cycle has elapsed, counting stops. If trigger starts again, set MTxRUN<MTRUN> = "1".

10.7.15 Stopping Type

By setting "0" to MTxRUN<MTRUN>, outputs and timers stop according to MTxIGCR<IGSTP[1:0]> setting.

10.7.15.1 Counter Stops with Initial State Output

When MTxIGCR<IGSTP[1:0]> is set to "00", the counter immediately stops and MTOUT0x/1x output becomes an initial value set with MTxIGOCR<IGPOL[1:0]>.

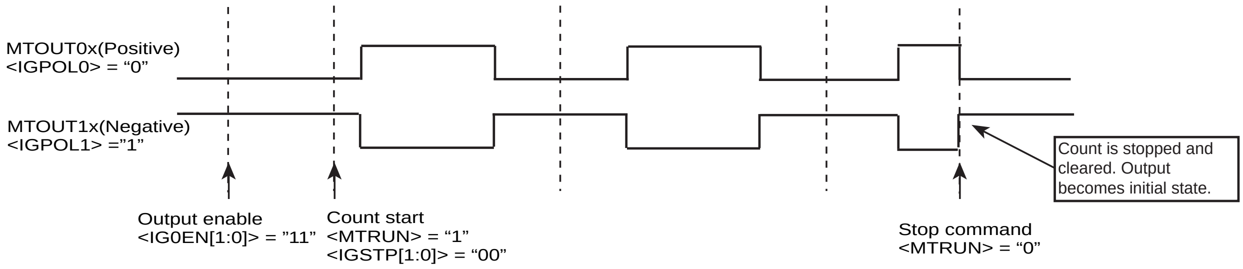


Figure 10-13 Counter stops with initial state output

10.7.15.2 Counter Stops with maintaining the output status

When <IGSTP[1:0]> is set to "01", the counter immediately stops and MTOUT0x/1x output is maintained.

If the counter starts again, set MTxRUN<MTRUN> = "1". At this time, outputs become an initial value (setting value of <IGPOL0> or <IGPOL1>) and then restarts.

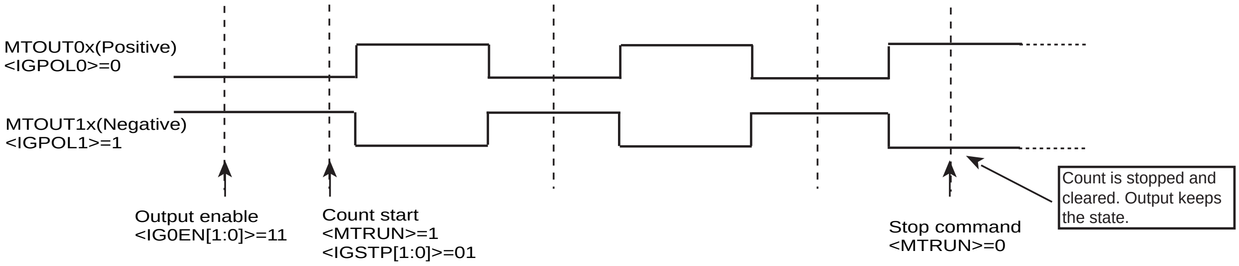


Figure 10-14 Counter stops with maintaining the output status

10.7.15.3 Counter Stops with Initial State after Cycle finished

When <IGSTP[1:0]> is set to "10", the counter operates until the cycle has finished. After cycle has finished, the counter stops. However, if trigger input becomes stop level until the cycle finishes, the counter stops at this point.

If the timer is set again, check if the counter stops after cycle has finished.

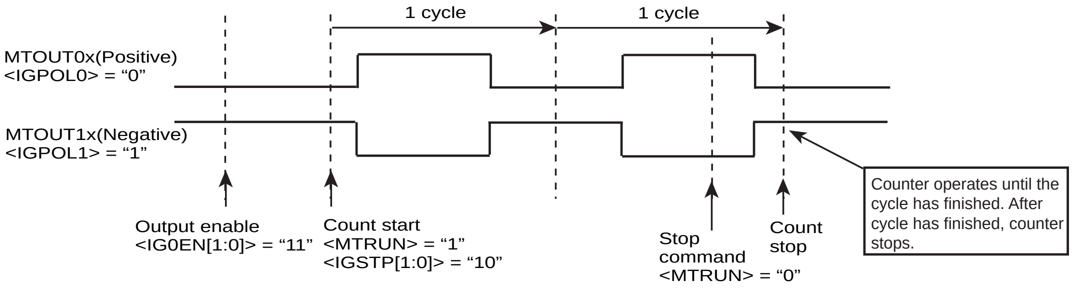


Figure 10-15 Counter stops with initial state after cycle has finished

10.7.16 Trigger Input

10.7.16.1 Logic of Trigger Input

The valid condition of MTxIN input is set with MTxIGICR<IGTRGSEL>.

- <IGTRGSEL> = "0" : Rising edge detection to start counting

  During "High" level, count-up is performed. During "Low" level, counter stops.
- <IGTRGSEL> = "1" : Falling edge detection to start counting

  During "Low" level, count-up is performed. During "High" level, counter stops.

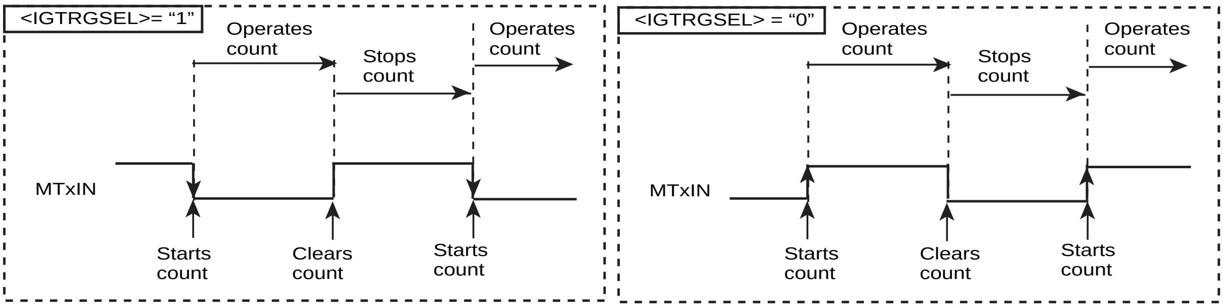


Figure 10-16 Logic of trigger input

While cycles are stopping, a stop trigger signal is accepted but a start signal is not. (Once a stop trigger signal is accepted while cycles are stopping, outputs become an initial value then the counter stops.)

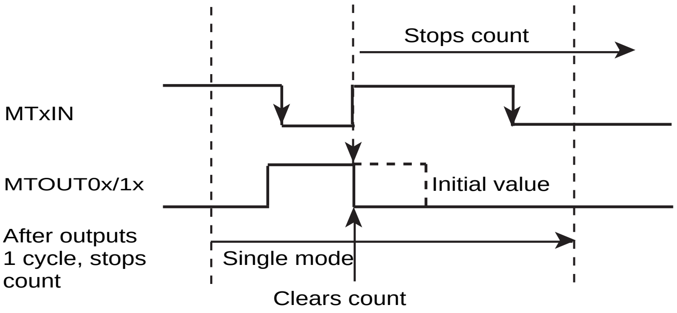


Figure 10-17 Trigger acceptance while cycles are stopping

10.7.16.2 Trigger Constant Acceptance/prohibit accessing during active level

MTxIGICR<IGTRGM> can choose either condition; one is a trigger from MTxIN is always accepted during PPG output, or another is a trigger is prohibited accessing during PPG output in active. This setting is only valid for enabled pin with MTxIGOCR <IGOEN[1:0]>.

When <IGTRGM> = "0" is set, a trigger input from MTxIN is always accepted regardless of MTOUT0x/1x in active/non-active. During this period, timer is started/stopped to clear and MTOUT0x/1x output becomes non-active.

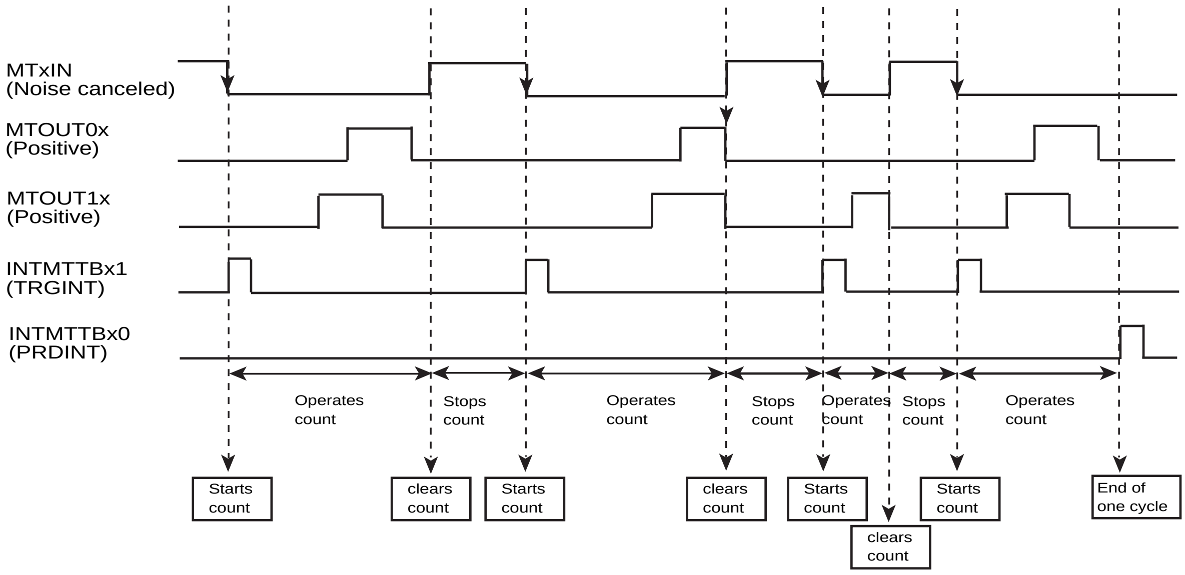


Figure 10-18 Trigger constant acceptance

When <IGTRGM> = "1" is set, input edge at MTOUT0x/1x output in non-active is accepted and cleared to stop.

If input edge at MTOUT0x/1x output in active, the counter does not immediately stops. It continues to count until MTOUT0x/1x output becomes non-active. When MTOUT0x/1x output is non-active, if trigger signal is not in active level, the counter is cleared to stop and waits next start trigger signal.

If the counter operates when both MTOUT0x and MTOUT1x are enabled, both outputs must be in non-active. Otherwise triggers are not accepted.

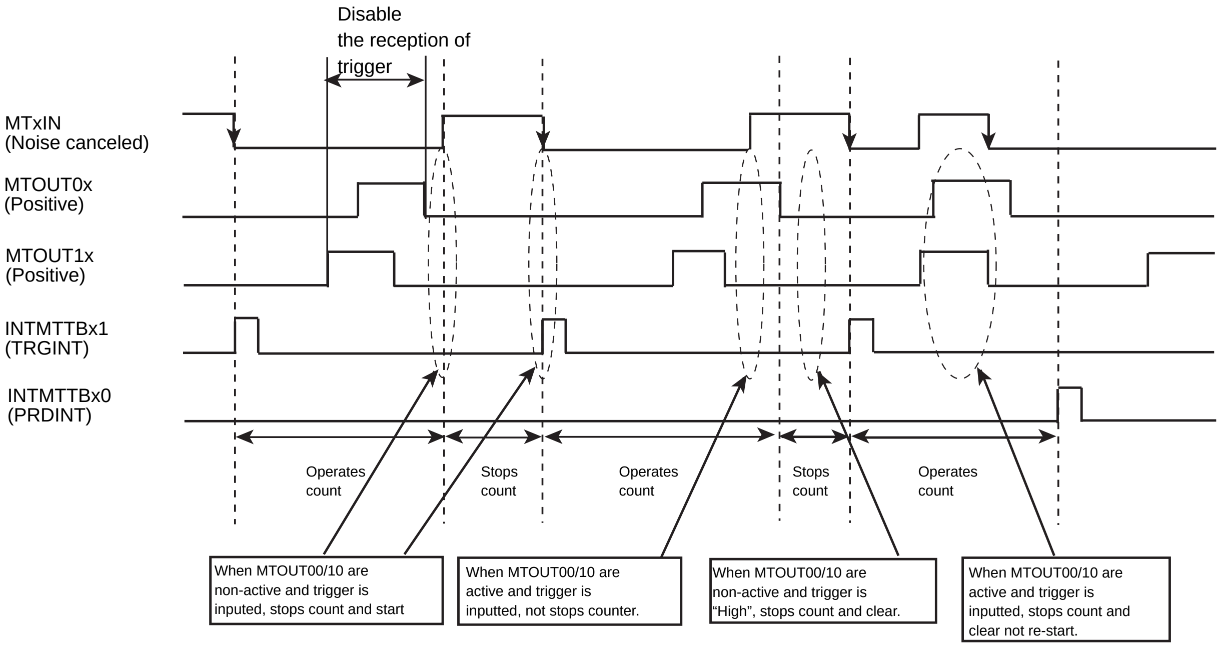

Disable
the reception of
trigger
MTxIN
(Noise canceled)
MTOUT0x
(Positive)
MTOUT1x
(Positive)
INTMTTBx1
(TRGINT)
INTMTTBx0
(PRDINT)
Operates count
Stops count
Operates count
Stops count
Operates count
When MTOUT00/10 are non-active and trigger is inputed, stops count and start
When MTOUT00/10 are active and trigger is inputted, not stops counter.
When MTOUT00/10 are non-active and trigger is "High", stops count and clear.
When MTOUT00/10 are active and trigger is inputted, stops count and clear not re-start.

Figure 10-19 Prohibit accessing during active level

10.7.17 Emergency Stop Function

10.7.17.1 Operation Description

When MTxIGEMGCR<IGEMGEN> = "1" is set, the emergency stop function is enabled (GEMGx pin is enabled to input).

If GEMGx pin detects a low level input, MTOUT0x/ MTOUT1x waveform becomes initial state (set with IGPOL0/IGPOL1) according to MTxIGEMGCR<IGEMGOC> setting or becomes high-impedance and generates a GEMGx interrupt.

This function prohibits only MTOUT0x/ MTOUT1x output. The counter does not stop so that timer must be stopped in the GEMG interrupt service routine.

10.7.17.2 Emergency stop monitor

On the emergency stop condition, MTxIGEMGST<IGEMGST> is set to "1". When IGEMGST is read, "1" indicates of emergency stopping.

10.7.17.3 GEMG interrupts

When an emergency stop input is received, a GEMG interrupt (INTMTEMGx) occurs. If this process uses interrupt service routine, the INTMTEMGx interrupt must be enabled in advance.

If GEMGx pin is "Low" and exits emergency stop status, GEMG interrupt occurs again, and MCU is in emergency stop condition again.

10.7.17.4 Exiting Emergency Stop Condition

When MCU exits emergency stop condition, check if GEMGx input is high and MTxRUN<MTRUN> is set to "0". Then confirm the timer stops (MTxIGST<IGST> = "0"), later MTxIGEMGCR<IGEMGRS> = "1" is set for exiting emergency stop condition.

When MTxIGCR<IGSTP[1:0]> = "01" or "10" is set in the stopping type selection register, set the initial setting with MTxIGOCR<IGPOL[1:0]> before writing MTxIGEMGCR<IGEMGRS> = "1".

10.7.18 Noise Canceller

The digital noise canceller eliminate noise inputting to external input pins (MTxIN and GEMGx).

It can be chosen the noise elimination time with MTxIGICR<IGNCSEL[3:0]> or MTxIGEMGCR <IGEMGCNT[3:0]> setting.

10.8 Operation Description of Motor Control Circuit (PMD : Programmable Motor Driver)

TMPM3U6FY/FW has two channels of motor control circuits (PMD).

The PMD enables 1-shunt sensor-less motor control by adding current-carrying output control and DC over voltage detection input. It achieves associative motor control with AD converter.

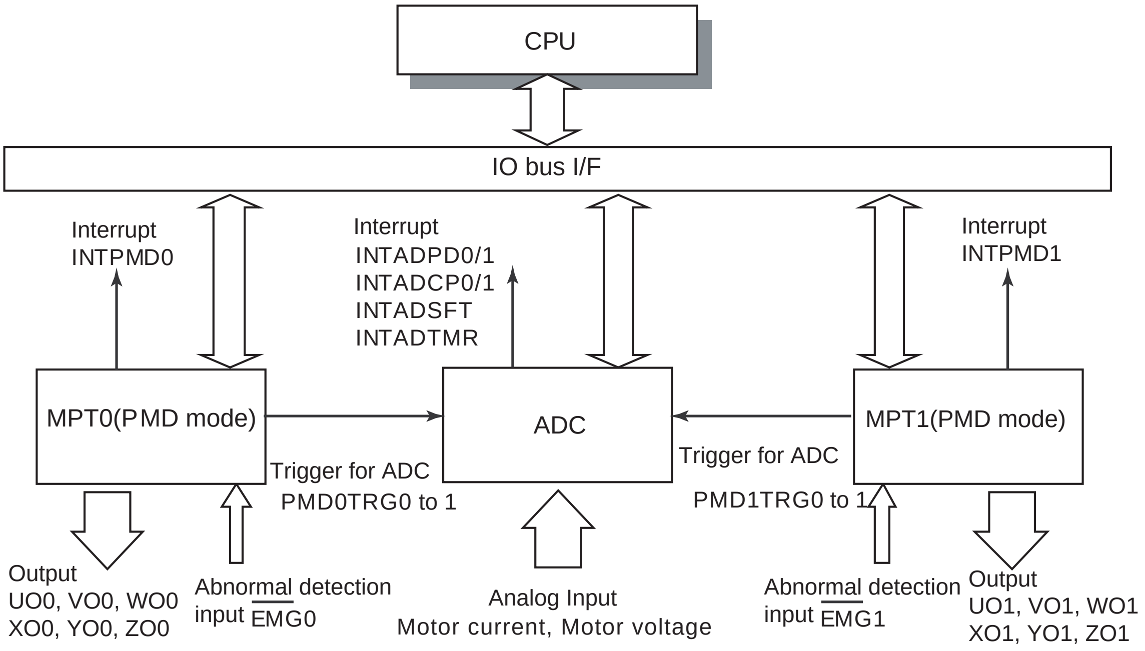


Figure 10-20 Block diagram of related with motor control

10.8.1 Input/output signal to PMD circuit

The following table describes input and output signals categorized by channel in motor control circuit.

Table 10-4 Input/output signals

CH	Pin name	PMD signal name	Function
PMD0	PC0/UO0	UO 0	U-phase output
	PC1/XO0	XO 0	X-phase output
	PC2/VO0	VO 0	V-phase output
	PC3YO0	YO 0	Y-phase output
	PC4/WO0	WO 0	W-phase output
	PC5/ZO0	ZO 0	Z-phase output
	PC6/$\overline{\text{EMG0}}$	EMG0	Abnormal detection input signal
PMD1	PG0/UO1	UO 1	U-phase output
	PG1/XO1	XO 1	X-phase output
	PG2/VO1	VO 1	V-phase output
	PG3/YO1	YO 1	Y-phase output
	PG4/WO1	WO 1	W-phase output
	PG5/ZO1	ZO 1	Z-phase output
	PG6/$\overline{\text{EMG1}}$	EMG1	Abnormal detection input signal

10.8.2 Structure

The PMD (Programmable Motor Driver) circuit consists of the wave generation circuit and synchronous trigger generation circuit. The wave generation circuit includes the pulse width modulation circuit, current-applying control circuit, protection control circuit and dead time control circuit.

- Pulse width modulation circuit generates identical 3-phase independent PWM waveforms.
- Applying current circuit determines each upper/lower output pattern of U-, V-, W-phase.
- Protection control circuit takes place emergency stop by detecting abnormal detect input.
- Dead time control prevent a short circuit at switching upper/lower phase.
- The synchronous trigger generation circuit generates synchronous trigger signals to the AD converter.

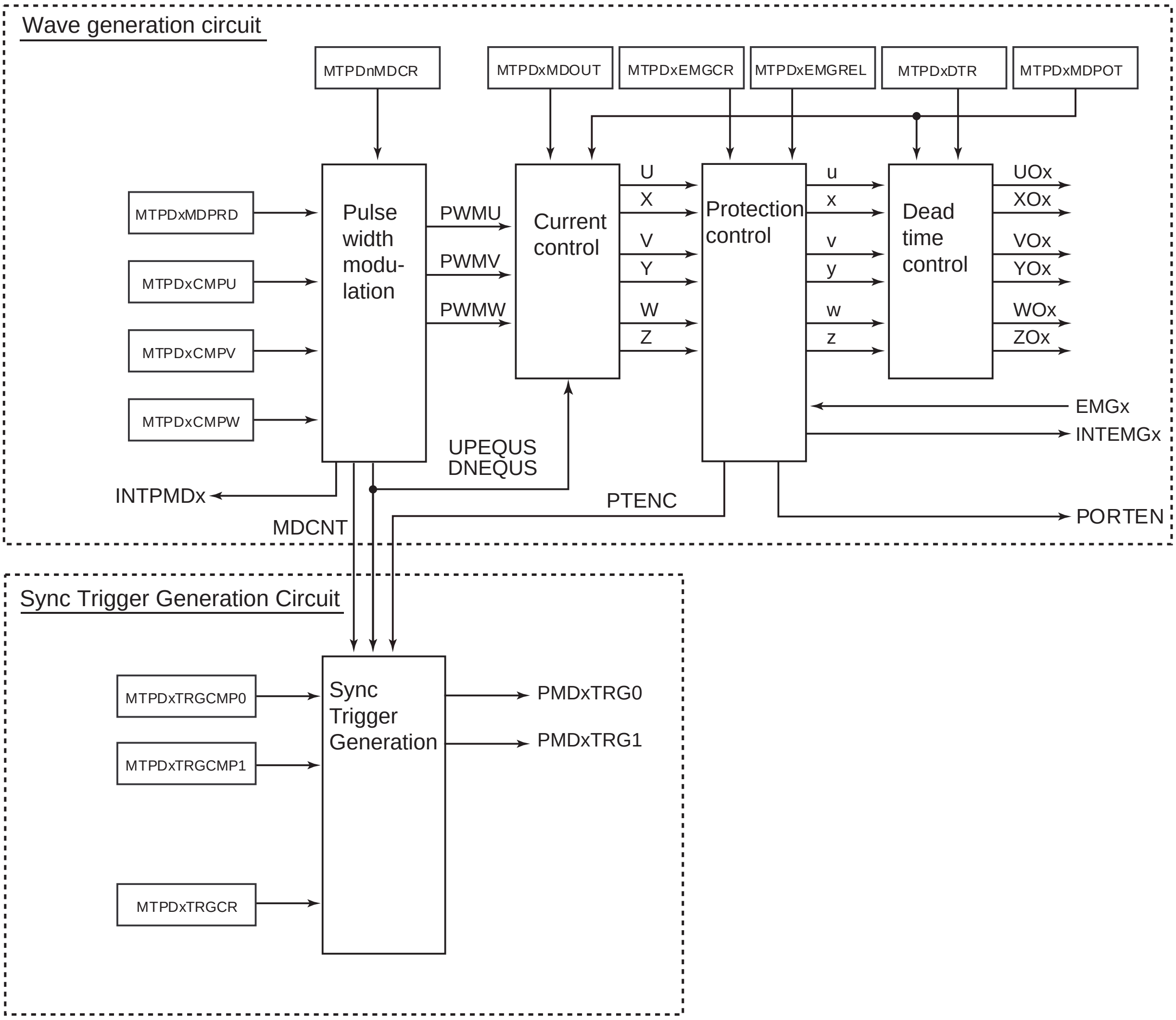


Figure 10-21 Schematic diagram of PMD circuit

10.8.3 Registers

10.8.3.1 Registers categorized by channel

This section describes registers and addresses of each channel.

Channel x	Base Address
Channel 0	0x4005_0400
Channel 1	0x4005_0480

Base address = 0x4005_0400

Register name (x = 0 to 1)		Address (Base+)
PMD enable register	MTPDxMDEN	0x0000
Port output mode register	MTPDxPORTMD	0x0004
PMD control register	MTPDxMDCR	0x0008
PWM count status register	MTPDxCNTSTA	0x000C
PWM count register	MTPDxMDCNT	0x0010
PWM cycle register	MTPDxMDPRD	0x0014
PWM compare U register	MTPDxCMPU	0x0018
PWM compare V register	MTPDxCMPV	0x001C
PWM compare W register	MTPDxCMPW	0x0020
Reserved	−	0x0024
PMD output control register	MTPDxMDOUT	0x0028
PMD output setting register	MTPDxMDPOT	0x002C
EMG release register	MTPDxEMGREL	0x0030
EMG control register	MTPDxEMGCR	0x0034
EMG status register	MTPDxEMGSTA	0x0038
Reserved	−	0x003C
Reserved	−	0x0040
Dead time register	MTPDxDTR	0x0044
Trigger Compare 0 Register	MTPDxTRGCMP0	0x0048
Trigger Compare 1 Register	MTPDxTRGCMP1	0x004C
Reserved	−	0x0050
Reserved	−	0x0054
Trigger Control Register	MTPDxTRGCR	0x0058
Trigger Output Mode Setting Register	MTPDxTRGMD	0x005C
Reserved	−	0x0060
Reserved	−	0x007C

10.8.3.2 MTPDxMDEN (PMD enable register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	PWMEN	R/W	Controls waveform generation to enable/disable 0: Disabled 1: Enabled While ports are set to PWM output, if <PWMEN> = "0" (disable) is set, output ports become high-impedance. Initial settings other than <PWMEN> such as output port polarity must be done before <PWMEN> = "1" (Enable) is set.

10.8.3.3 MTPDxPORTMD (Port output mode register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	-	R/W	Write "0".
0	PORTMD	R/W	Sets port control 0: High-impedance 1: PMD output Six output ports with all phase output control signals to external port by setting <PORTMD>. If tool break occurs when high-impedance is chosen, external output port becomes high-impedance. Otherwise PMD output is set. Note 1) When MTPDxMDEN<PWMEN> = "0" is set, high-impedance output is set regardless of output port setting. Note 2) External port output control can be done according to PMDxEMGMD setting, even when EMG input.

10.8.3.4 MTPDxMDCR (PMD control register)

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

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as "0".
6	PWMCK	R/W	Specifies PWM cycle extension mode 0: Normal cycle 1: 4-fold cycle In the normal setting, PWM counter operates at a resolution of 25ns@fsys = 40MHz. Note) Sawtooth wave 25ns, Triangle wave 50ns In the 4-fold cycle setting, PWM counter operates at a resolution of 100ns@2-bit counter (fsys = 40MHz) Note) sawtooth wave 100ns, Triangle wave 200ns
5	SYNTMD	R/W	Sets port output of U-, V- and W-phases. Note) Refer toTable 10-6.
4	DTYMD	R/W	Chooses DUTY mode 0: U-phase in common 1: 3-phase independent Chooses duty setting among either each 3-phase (PMDxCMPU, V and W) is independent or PMDxCMPU register in each 3-phase is used in common.
3	PINT	R/W	Chooses PWM interrupt timing when PWM mode 1 (triangle wave) is set. 0: When PWM count MDCNT = "1" is set, (minimum) interrupt request occurs. 1:When PWM count MDCNT = MTPDxMDPRD<MDPRD> is set, (maximum) interrupt request occurs. User can be choose the interrupt generation timing in the PWM mode 1 (triangle wave) either when PWM counter MDCNT becomes "1" (minimum) or when PWM counter becomes MTPDxMDPRD<MDPRD> (maximum). If PWM interrupt cycle is set as 0.5 cycle with <INTPRD>, PWM interrupt occurs both when PWM counter MDCNT becomes "1" (minimum) and <MDPRD> (maximum) regardless of this register. Also in PWM mode 0 (sawtooth wave), PWM interrupt occurs when PWM counter MDCNT becomes <MDPRD> (maximum) regardless of this register.
2-1	INTPRD	R/W	Chooses PWM interrupt cycle 00: Every PWM 0.5 cycles (can be set in PWM mode1 (triangle wave) 01: Every PWM 1 cycle 10: Every PWM 2 cycles 11: Every PWM 4 cycles Frequency of PWM interrupt generation can be chosen among every 0.5 cycles, 1 cycles, 2 cycles or 4 cycles.
0	PWMMD	R/W	Specifies PWM carrier wave 0: PWM mode 0 (edge PWM, sawtooth wave) 1: PWM mode 1 (center PWM, triangle wave)

10.8.3.5 MTPDxCNTSTA (PWM count status register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as "0".
0	UPDWN	R	PWM counter flag 0: Up-counting 1: Down-counting Indicate PWM counter is up-counting or down-counting. When PWM mode 0 (sawtooth wave) is chosen, always read "0".

10.8.3.6 MTPDxMDCNT (PWM count register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MDCNT[15:0]	R	16-bit read-only register for counting PWM cycles. When MTPDxMDCR<PWMCK> = "0" is set, PWM counter resolution is 25ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 50ns@fsys = 40MHz in PWM mode 1 (triangle wave). When <PWMCK> = "1" is set, PWM counter resolution is 100ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 200ns@fsys = 40MHz in PWM mode 1 (triangle).

10.8.3.7 MTPDxMDPRD (PWM cycle register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	MDPRD[15:0]	R/W	Sets PWM cycles. PWM counter resolution is 25ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 50ns@fsys = 40MHz in PWM mode 1 (triangle wave). When MTPDxMDCR<PWMCK> = "1" is set, PWM counter resolution is 100ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 200ns@fsys = 40MHz in PWM mode 1 (triangle wave). <MDPRD> is a PWM cycle setting register with double-buffering structure. Thus even if PWM counter is operating, it can be changed. Transfer timing from register to latch circuit is when PWM counter MDCNT matches with MTPDxMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDxMCR<INTPRD> = "00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Sets a value of 0x10 or more to <MDPRD>. Even if a value less than 0x10 is set, the register is set up as 0x10. (User specified value is set in the register.) Read a value of register (data set from bus) when read.

Note: **Use half-word access or word access.**

10.8.3.8 MTPDxCMPU (PWM compare register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	CMPU[15:0]	R/W	Controls PWM pulse width Resolutions are 25ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) and 50ns@fsys = 40MHz in PWM mode 1 (triangle wave). When MTPDxMDCR<PWMCK> is set to "1", a resolution is set to 100ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 200ns@fsys = 40MHz in PWM mode 1 (triangle). <CMPU> is the compare register that determines the pulse width outputting to U-phase. This register compares a pulse large or small with PWM counter MDCNT to determine the pulse width. <CMPU> is double-buffering structure, so that even if PWM counter is operating, it can be changed. Transfer timing from buffers to registers is when PWM counter MDCNT matches with MTPDxMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDxMCR <INTPRD> = "00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Read a value of buffer (data set from bus) when read.

Note: **Use half-word access or word access.**

10.8.3.9 MTPDxCMPV (PWM compare register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	CMPV[15:0]	R/W	Sets PWM pulse width Resolutions are 25ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) and 50ns@fsys = 40MHz in PWM mode 1 (triangle wave). When MTPDxMDCR<PWMCK> is set to "1", a resolution is set to 100ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 200ns@fsys = 40MHz in PWM mode 1 (triangle). <CMPV> is the compare register that determines the pulse width outputting to U-phase. This register compares a pulse large or small with PWM counter MDCNT to determine the pulse width. <CMPV> is double-buffering structure, so that even if PWM counter is operating, it can be changed. Transfer timing from buffers to registers is when PWM counter MDCNT matches with MTPDxMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDxMCR <INTPRD> = "00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Read a value of buffer (data set from bus) when read.

Note: **Use half-word access or word access.**

10.8.3.10 MTPDxCMPW (PWM compare register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-0	CMPW[15:0]	R/W	Sets PWM pulse width Resolutions are 25ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) and 50ns@fsys = 40MHz in PWM mode 1 (triangle wave). When MTPDxMDCR<PWMCK> is set to "1", a resolution is set to 100ns@fsys = 40MHz in PWM mode 0 (sawtooth wave) or 200ns@fsys = 40MHz in PWM mode 1 (triangle). <CMPW> is the compare register that determines the pulse width outputting to W-phase. This register compares a pulse large or small with PWM counter MDCNT to determine the pulse width. <CMPW> is double-buffering structure, so that even if PWM counter is operating, it can be changed. Transfer timing from buffers to registers is when PWM counter MDCNT matches with MTPDxMDPRD <MDPRD>. If interrupt timing is set to 0.5 cycles (MTPDxMCR <INTPRD> = "00"), the transfer is taken place when PWM counter MDCNT is set to "1" or matches with <MDPRD>. Read a value of buffer (data set from bus) when read.

Note: **Use half-word access or word access.**

10.8.3.11 MTPDxMDOUT (PMD output control register)

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

Bit	Bit Symbol	Type	Function
31-11	-	R	Read as "0".
10	WPWM	R/W	Controls U-, V- and W-phase outputs 0: H/L output 1: PWM output For details, refer to Table 10-6.
9	VPWN	R/W	
8	UPWN	R/W	
7-6	-	R	Read as "0".
5-4	WOC[1:0]	R/W	Controls U-, V- and W-phase outputs For details, refer to Table 10-6.
3-2	VOC[1:0]	R/W	
1-0	UOC[1:0]	R/W	

Note: **Use half-word access or word access.**

10.8.3.12 MTPDxMDPOT (PMD output setting register)

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

Bit	Bit Symbol	Type	Function
31-4	-	R	Read as "0".
3	POLH	R/W	Chooses polarity of upper phase output port 0: Low active 1: High active Note) Chooses the setting when MTPDxMDEN<PWMEN> = "0" is set.
2	POLL	R/W	Choose polarity of lower phase output port 0: Low active 1: High active Note) Chooses the setting when MTPDxMDEN<PWMEN> = "0" is set.
1-0	PSYNCS[1:0]	R/W	Chooses the timing when port outputs of U-, V- and W-phase output setting is reflected. 00: Reflects when write 01: Reflects when PWM counter MDCNT = "1" (minimum) 10: Reflects when PWM counter MDCNT = MTPDxMDPRD<MDPRD> (maximum) 11: Reflects when PWM counter MDCNT = "1"(minimum) and MTPDxMDPRD<MDPRD> (maximum) Note) Chooses the setting when MTPDxMDEN<PWMEN> = "0" is set.

10.8.3.13 MTPDxEMGREL (EMG release register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	EMGREL[7:0]	W	Writes EMG prohibition code To prohibit EMG function, set the procedure as follows; set "0x5A"→"0xA5" to <EMGREL[7:0]>, then set "0" to MTPDxEMGCR<EMGEN>.

Note: **To prohibit EMG function, three instructions must be executed consecutively. Three instructions are as follows; set "0x5A", change to "0xA5" and set "0" to MTPDxEMGCR<EMGEN>.**

10.8.3.14 MTPDxEMGCR (EMG control register)

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

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	EMGCNT[3:0]	R/W	Set the noise elimination time for abnormal condition detection input Noise elimination time is calculated with the following formula: EMGCNT[3:0]×16/fsys 0000: Noise filter is not used. 0001: Input noise elimination time 16 / fsys [s] 0010: Input noise elimination time 32 / fsys [s] 0011: Input noise elimination time 48 / fsys [s] 0100: Input noise elimination time 64 / fsys [s] 0101: Input noise elimination time 80 / fsys [s] 0110: Input noise elimination time 96 / fsys [s] 0111: Input noise elimination time 112 / fsys [s] 1000: Input noise elimination time 128 / fsys [s] 1001: Input noise elimination time 144 / fsys [s] 1010: Input noise elimination time 160 / fsys [s] 1011: Input noise elimination time 176 / fsys [s] 1100: Input noise elimination time 192 / fsys [s] 1101: Input noise elimination time 208 / fsys [s] 1110: Input noise elimination time 224 / fsys [s] 1111: Input noise elimination time 240 / fsys [s]
7-6	-	R	Read as "0".
5	INHEN	R/W	Chooses a PMD output status at tool break 0: PMD output is continued 1: High-impedance Initial state is high-impedance.
4-3	EMGMD	R/W	EMG protection mode selection 00: All phase are on/PORT output and high-impedance 01: All phase are off/PORT output and high-impedance 10: All phase are on/PORT output is enabled. 11: All phase are off/PORT output is high-impedance. Note) On = PWM output (No output control) Off = Low (@high-active (POLL/H = 1)) Upon EMG occurred, this bit controls that five PWM outputs in all phase (upper and lower) are On/Off. Also, it controls that PORT output is disabled/enabled when EMG occurred.
2	-	R/W	Write "0".
1	EMGRS	R/W	Returns from EMG protection status 0: − 1: Returns from protection status When after MTPDxMDOUT<WPWM><VPWM><UPWM><WOC[1:0]><VOC[1:0]><UOC[1:0]> are set to 0, MCU returns from EMG protection status by setting <EMGRS> to "1". Always reads as "0".
0	EMGEN	R/W	Sets EMG protection circuit to disable/enable 0: Disabled 1: Enabled When this bit is set to "1", EMG protection circuit operates. Initial state is enabled. To disable EMG protection circuit, set as follows; set "0x5A"→"0xA5 to "MTPDxEMGREL<EMGREL> in order, then set "0" to <EMGEN>.(These three instruction must be executed consecutively.)

Note: **When returning from EMG protection status by setting MTPDxEMG<EMGRS>, read MTPDxEMGSTA<EMGI> to confirm if abnormal detection input level is "H".**

10.8.3.15 MTPDxEMGSTA (EMG status register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	EMGI	R	Monitors the level of abnormal condition input 0: Abnormal condition input level is "L" 1: Abnormal condition input level is "H"
0	EMGST	R	Monitors EMG protection condition 0: Normal operation 1: During in EMG protection

10.8.3.16 MTPDxDTR (Dead Time Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-0	DTR[7:0]	R/W	Sets dead time Dead time is calculated with the following formula: 200ns x <DTR[7:0]> (fsys = 40MHz)

Note: **Do not modify MTPDxDTR<DTR[7:0]> when MTPDxMDEN<PWMEN> = "1" is set.**

10.8.3.17 MTPDxTRGCMP0(Trigger Compare 0 Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-16</td><td>-</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>15-0</td><td>TRGCMP0[15:0]</td><td>R/W</td><td><TRGCMP0> outputs PMDTRG when <TRGCMP0> matches with MTPDxMDCNT<MDCNT>.
If "0x0001" is set, trigger is output after PWM starts (MTPDxMDEN<PWMEN> = "1") in the PWM mode 0 (sawtooth wave) except only at first one cycle.
When read, a buffer value is read (data set in the bus).

<TRGCMP0> is double-buffered structure. A trigger compare register0 update timing varies on mode setting of MTPDxTRGCR<TRG0MD>. If MTPDxTRGCR<TRGBE> = "1" is set, the trigger compare register0 is always updated regardless of trigger mode.

Trigger output mode setting and update timing of trigger compare register0
<table>
<tr><th>MTPDxTRGCR<TRGxMD></th><th>Update timing</th></tr>
<tr><td>000 : Trigger is disabled.</td><td>Always updated</td></tr>
<tr><td>001 : A match when down-counting</td><td>Register is updated at the peak of PWM carrier (when UC matches with MTPDxMDPRD<MDPRD>).</td></tr>
<tr><td>010 : A match when up-counting</td><td>Register is updated at the bottom of PWM carrier (when UC matches with 1).</td></tr>
<tr><td>011 : A match when up-/down-counting</td><td>Register is updated at the peak/bottom of PWM carrier.</td></tr>
<tr><td>100 : PWM carrier peak</td><td rowspan="4">Always updated</td></tr>
<tr><td>101 : PWM carrier bottom</td></tr>
<tr><td>110 : PWM carrier peak or bottom</td></tr>
<tr><td>111 : Trigger is disabled.</td></tr>
</table></td></tr>
</table>

Note 1: **Use half-word access or word access.**

Note 2: **Set 1 ≤ <TRGCMP0> ≤ (MTPDxMDPRD<MDPRD>-1).**

10.8.3.18 MTPDxTRGCMP1(Trigger Compare 1 Register)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-16</td><td>-</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>15-0</td><td>TRGCMP1[15:0]</td><td>R/W</td><td>&lt;TRGCMP1> outputs PMDTRG when <TRGCMP1> matches with MTPDxMDCNT<MDCNT>.
If "0x0001" is set, trigger is output after PWM starts (MTPDxMDEN<PWMEN> = "1") in the PWM mode 0 (sawtooth wave) except only at first one cycle.
When read, a buffer value is read (data set in the bus).

<TRGCMP1> is double-buffered structure. A trigger compare register1 update timing varies on mode setting of MTPDxTRGCR<TRG1MD>. If MTPDxTRGCR<TRGBE> = "1" is set, the trigger compare register1 is always updated regardless of trigger mode.

Trigger output mode setting and update timing of trigger compare register1
<table>
<tr><th>MTPDxTRGCR<TRGxMD&gt;</th><th>Update timing</th></tr>
<tr><td>000 : Trigger is disabled.</td><td>Always updated</td></tr>
<tr><td>001 : A match when down-counting</td><td>Register is updated at the peak of PWM carrier (when UC matches with MTPDxMDPRD<MDPRD&gt;).</td></tr>
<tr><td>010 : A match when up-counting</td><td>Register is updated at the bottom of PWM carrier (when UC matches with 1).</td></tr>
<tr><td>011 : A match when up-/down-counting</td><td>Register is updated at the peak/bottom of PWM carrier.</td></tr>
<tr><td>100 : PWM carrier peak</td><td rowspan="4">Always updated</td></tr>
<tr><td>101 : PWM carrier bottom</td></tr>
<tr><td>110 : PWM carrier peak or bottom</td></tr>
<tr><td>111 : Trigger is disabled.</td></tr>
</table>
</td></tr>
</table>

Note 1: **Use half-word access or word access.**

Note 2: **Set 1 ≤ <TRGCMP1> ≤ (MTPDxMDPRD<MDPRD>-1).**

10.8.3.19 MTPDxTRGCR (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	-	-	-	-	-	-	-	-
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".
15-8	-	R/W	Write "0".
7	TRG1BE	R/W	Update timing setting of trigger compare register1 (PMDxTRG1) 0: Synchronous update 1: Asynchronous update (Asynchronous update of buffer is enabled. A value is immediately reflected after write.)
6-4	TRG1MD[2:0]	R/W	Mode setting of PMDxTRG1 000: Trigger output is disabled. 001: Trigger output when matching in down-counting 010: Trigger output when matching in up-counting 011: Trigger output when up-/down-counting 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 is disabled. Selects the trigger output match mode. If the edge mode is selected in PMD, up-count and carrier peak output is selected even if down-count/carrier bottom is selected. If "011" is selected, a trigger output is performed once per one cycle in the PWM mode 1 (triangle wave) while TRGCMP = 0x0001 is set.
3	TRG0BE	R/W	Update timing setting of trigger compare register0 (PMDxTRG0) 0: Synchronous update 1: Asynchronous update (Asynchronous update of buffer is enabled. A value is immediately reflected after write.)
2-0	TRG0MD[2:0]	R/W	Mode setting of PMDxTRG0 000: Trigger output is disabled. 001: Trigger output when matching in down-counting 010: Trigger output when matching in up-counting 011: Trigger output when up-/down-counting 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 is disabled. Selects the trigger output match mode. If the edge mode is selected in PMD, up-count/carrier peak output is selected even if down-count/carrier bottom is selected. If "011" is selected, a trigger output is performed once per one cycle in the PWM mode 1 (triangle wave) while TRGCMP = 0x0001 is set.

10.8.3.20 MTPDxTRGMD (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	-	-	-	-	-	-	-	EMGTGE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1	-	R/W	Write "0".
0	EMGTGE	R/W	Trigger output enable setting during EMG protection in operation. 0: Trigger output is disabled when protection is operating. 1: Trigger output is enabled when protection is operating.

10.9 Operation Description categorized by Circuit

10.9.1 Pulse Width Modulation Circuit

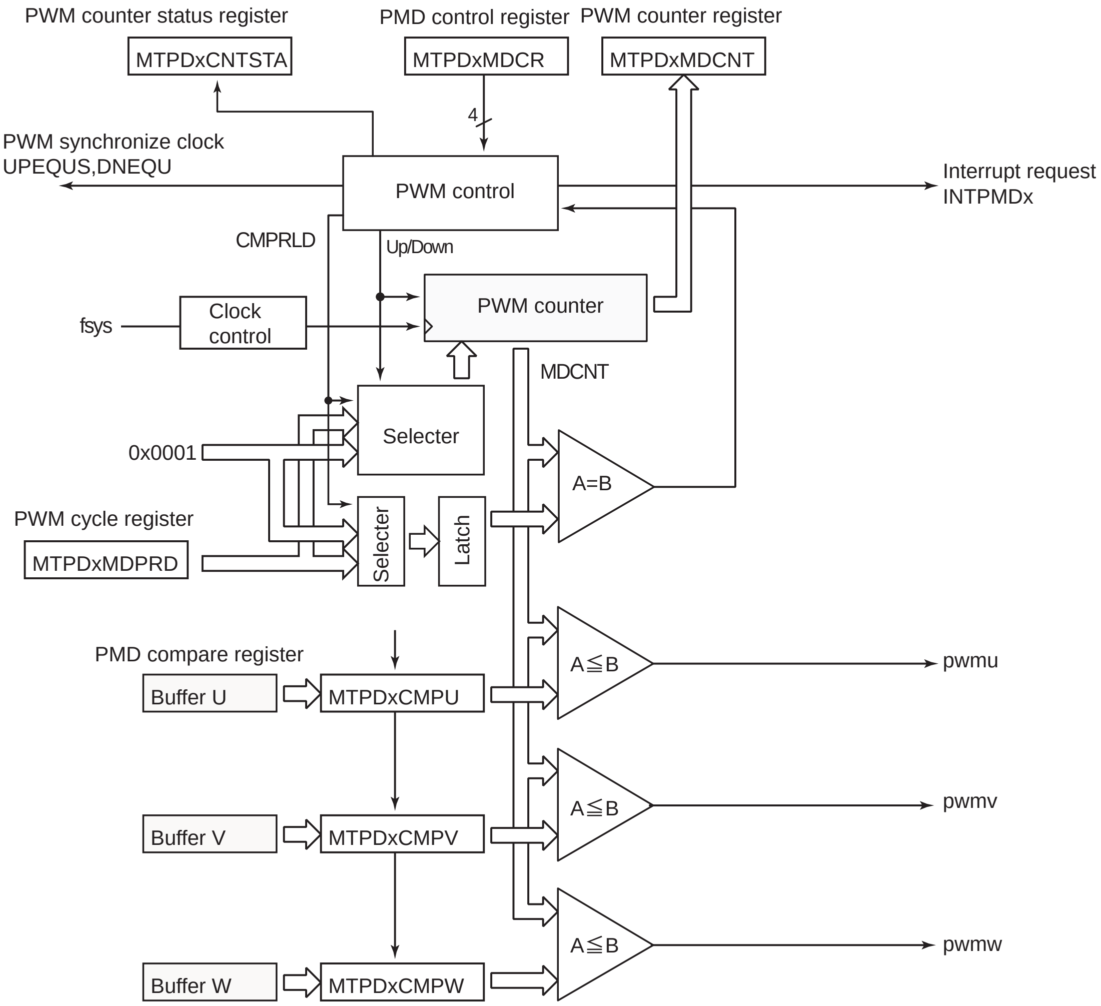


Figure 10-22 Schematic diagram of pulse width modulation circuit

Pulse width modulation circuit contains PWM counter MDCNT which is a 16-bit up-/down-counter. In PWM mode 0 (sawtooth wave), the counter resolution is 25ns@fsys = 40MHz. In PWM mode 1(triangle wave), the counter resolution is 50ns@fsys = 40MHz to generate PWM carrier. When MTPDxMDCR<PWMCK> = "1" is set, the counter resolution is 100ns@fsys = 40MHz in PWM mode 0 (sawtooth wave). In the PWM mode 1 (triangle wave), the counter resolution is 200ns@fsys = 40MHz to generate PWM carrier.

PWM carrier waveform mode can be chosen either an edge PWM (sawtooth wave) in PWM mode 0 or a center PWM (triangle wave) in mode 1.

1. PWM cycle setting

   MTPDxMDPRD<MDPRD> register determines PWM cycles.

   MTPDxMDPRD register contains a latch circuit with double-buffering structure. Register values are synchronously transferred as the comparator input (latch) when PWM counter MDCNT matches with <MDPRD>. If MTPDxMDCR<INTPRD> is set to "00", updating on every PWM half cycle can be chosen.

$$\text{Sawtooth wave PWM : Value of MDPRD} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]}}$$

$$\text{Triangle wave modulation PWM : Value of MDPRD register} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz] x 2}}$$

2. Compare function

   This compare function generates desirable duty PWM waveforms by small/large comparing a value of 3-phase PWM compare register (PMDxCMPU/V/W) with a carrier generated by PWM counter MDCNT.

   PWM compare registers in each phase have double-buffering structure. Buffer values are synchronously transferred to PWM compare register when internal counter value matches with <MDPRD>. If MTPDxMDCR<INTPRD> is set to "00", loading on every PWM half cycle can be chosen.

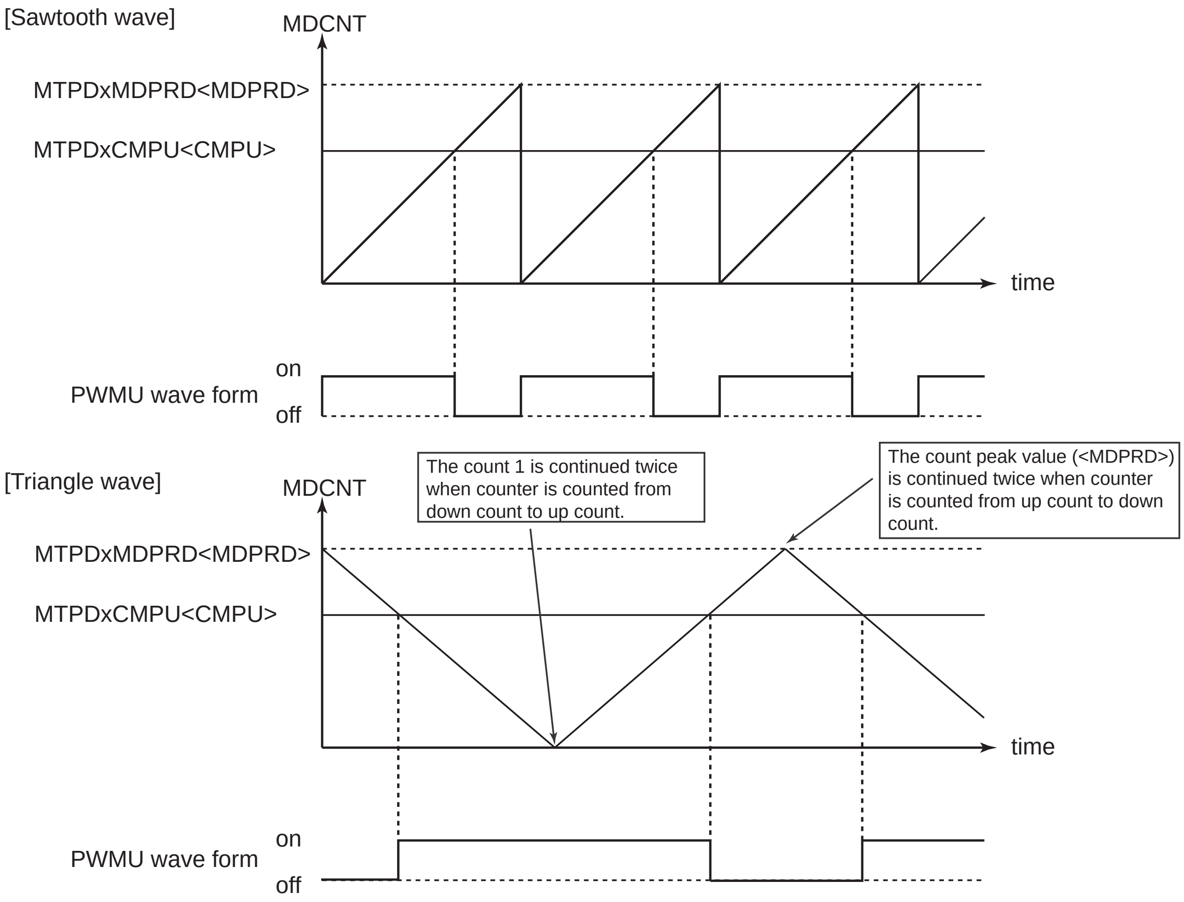

[Sawtooth wave]
MDCNT
MTPDxMDPRD<MDPRD>
MTPDxCMPU<CMPU>
time
PWMU wave form
on
off
[Triangle wave]
MDCNT
The count 1 is continued twice when counter is counted from down count to up count.
The count peak value (<MDPRD>) is continued twice when counter is counted from up count to down count.
MTPDxMDPRD<MDPRD>
MTPDxCMPU<CMPU>
time
PWMU wave form
on
off

Figure 10-23 PWM waveform

3. Waveform mode

   Two kinds of 3-phase PWM generation can be chosen.

   1. 3-phase independent duty mode: Set independent values to each 3-phase compare register to generate 3-phase independent PWM waveforms. This is used for generating arbitrary drive waveform such as a sine wave.

   2. 3-phase common duty mode: Set only U-phase PWM compare register. By setting a value in U-phase, identical PWM waveform in 3-phase common. This is used for square waveform drive used in DC motor.

4. Interrupt service routine

   In pulse width modulation circuit, PWM interrupts synchronously occurs with PWM waveforms. A frequency of PWM interrupt is chosen among once every half PWM cycle, once every one PWM cycle, once every two PWM cycles or once every four PWM cycles.

10.9.2 Applying Current Control Circuit

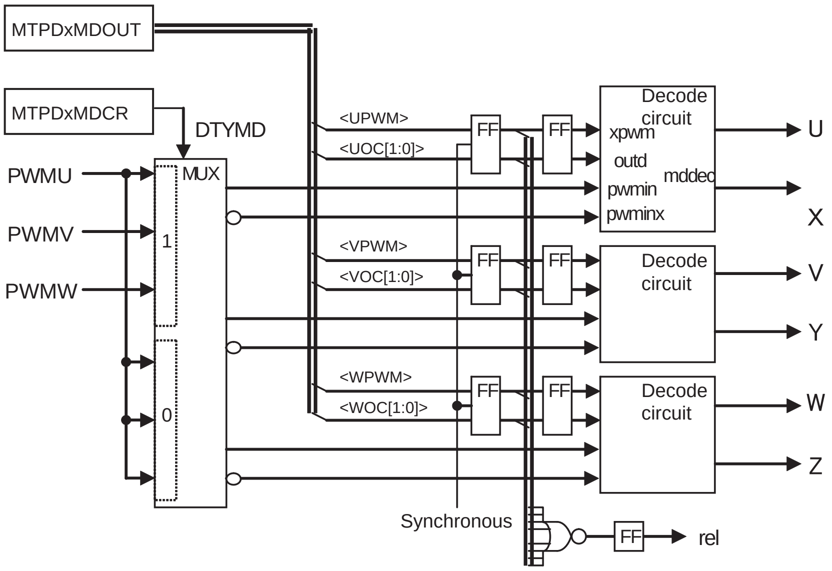


Figure 10-24 Schematic diagram of applying current control circuit

This circuit controls output port according to the content set in the PMD output register (MTPDxMDOUT). The settings can be divided two contents such as the selection of synchronous signal at port output, and the port output setting. The port output setting is double-buffering structure, so that the update timing can be chosen between synchronous update or asynchronous update with PWM.

The output setting of six ports can be set independently between active or inactive using MTPDxMDPOT<POLH> <POLL>. In addition, PWM output or H/L output can be chosen in each three phase (U-, V- and W-phase) using MTPDxMDOUT<WPWM><VPWM><UPWM>. If PWM output is chosen, PWM waveform generates. If H/L output is chosen, fixed-high output or fixed-low output generates. For the relation between the port output settings using MTPDxMDOUT and the pin output using the polarity of PMD control register (MTPDxMDCR), refer to"Table 10-6 Port outputs setting by UOC, VOC, WOC, UPWN, VPWN and WPWM bits".

Also, one shunt current can be detected as follows:

Table 10-5 Settings for one shunt current detection

	Normal	U-phase PWM shift	V-phase PWM shift	W-phase PWM shift
CMPU	duty_U	MTPDxMDPRD <MDPRD>-duty_U	duty_U	duty_U
CMPV	duty_V	duty_V	MTPDxMDPRD <MDPRD>D-duty_V	duty_V
CMPW	duty_W	duty_W	duty_W	MTPDxMDPRD <MDPRD>-duty_W
<UOC>	11	00	11	11
<VOC>	11	11	00	11
<WOC>	11	11	11	00

Table 10-6 Port outputs setting by UOC, VOC, WOC, UPWN, VPWN and WPWM bits

MTPDxMDCR<SYNTMD> = "0"

Polarity: high-active (MTPDxMDPOT<POLH><POLL> = "11")

MDOUT output control		MTPDxMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

MTPDxMDCR<SYNTMD> = "0"

Polarity: low-active (MTPDxMDPOT<POLH><POLL> = "00")

MDOUT output control		MTPDxMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	$\overline{\text{PWM}}$
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

MTPDxMDCR<SYNTMD> = "1"

Polarity: high-active (MTPDxMDPOT<POLH><POLL> = "11")

MDOUT output control		MTPDxMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	$\overline{\text{PWM}}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

MTPDxMDCR<SYNTMD> = "1"

Polarity: low-active (MTPDxMDPOT<POLH><POLL> = "00")

MDOUT output control		MTPDxMDOUT <WPWM><VPWM><UPWM> H/L/PWM output selection			
<WOC[1]> <VOC[1]> <UOC[1]> (Upper)	<WOC[0]> <VOC[0] ><UOC[0]> (Lower)	0 : H/L output		1 : PWM output	
		Upper output	Lower output	Upper output	Lower output
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

10.9.3 Protection Control Circuit

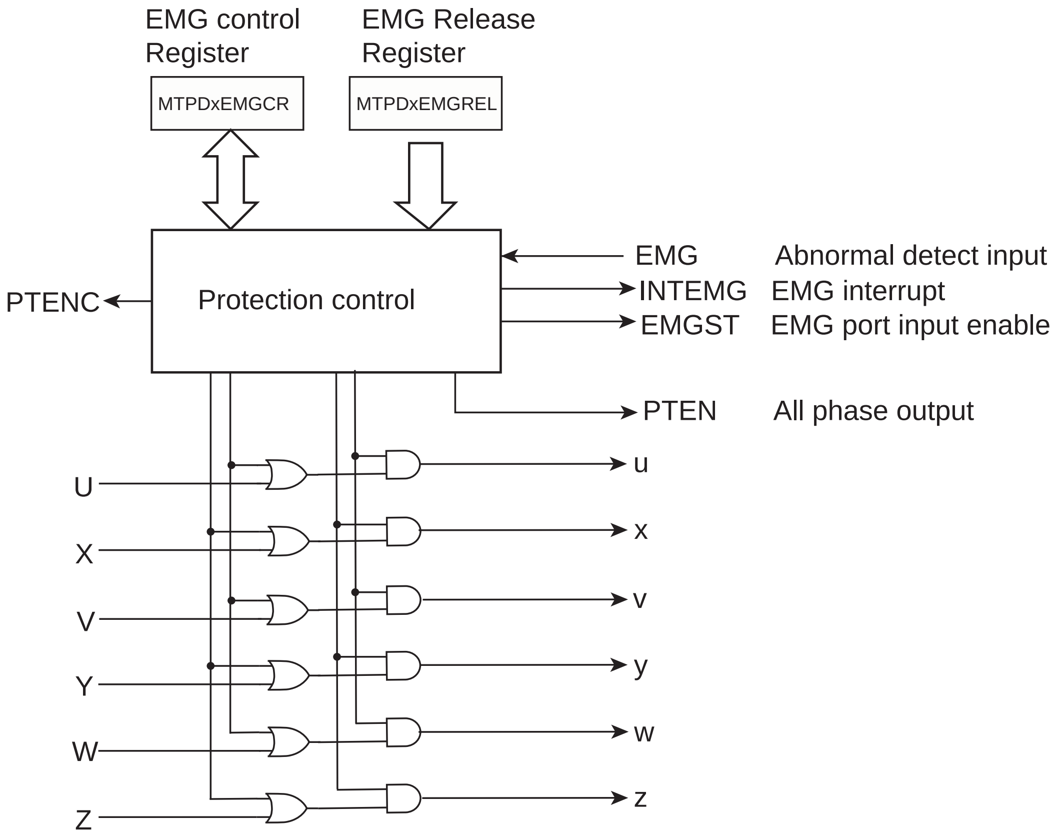


Figure 10-25 Schematic diagram of protection control circuit

The protection control circuit consists of the protection control part and the port output prohibition circuit part. It operates when abnormal detection input is low level. The EMG protection circuit is used for emergency stop. If abnormal detection input is found (high level → low level), six PWM outputs are immediately prohibited (depending on MTPDxEMGCR<EMGMD> setting) and an EMG interrupt (INTEMG) occurs.

In addition, this circuit outputs control signals that become external output ports to be high-impedance by setting <EMGMD>.

This circuit prohibits six PWM outputs when PMD is stopped caused by tool break as well. This prohibition depends on <EMGMD> setting. At tool break, it can choose the high-impedance control of external output port by setting MTPDxPORTMD<PORTMD>.

When MTPDxEMGSTA<EMGST> reads as "1", this indicates MCU is in the EMG protection status.

To return from the EMG protection status, set as follows; all ports are set to in-active (MTPDxMDOUT <WPWM> <VPWM> <UPWM> <WOC[1:0]> <VOC[1:0]> <UOC[1:0]> = "0"); then MTPDxEMGCR <EMGRS> is set to "1".

To prohibit the EMG function, set as follows; set 0x5A and 0xA5 to the EMG prohibition code register (MTPDxEMGREL<EMGREL[7:0]>) in order; set "0" to MTPDxEEMGCR<EMGEN> (These three instruction must be executed consecutively.). However, the returning service routine is ignored while abnormal detection inputs are low. The returning service routine must be taken place when abnormal detection input level becomes high after confirming MTPDxEMGSTA<EMGI> is high.

By setting specified key codes (0x5A and 0xA5) to <EMGREL[7:0]>, the EMG protection circuit is enabled to prevent unintentional operation.

10.9.4 Dead Time Circuit

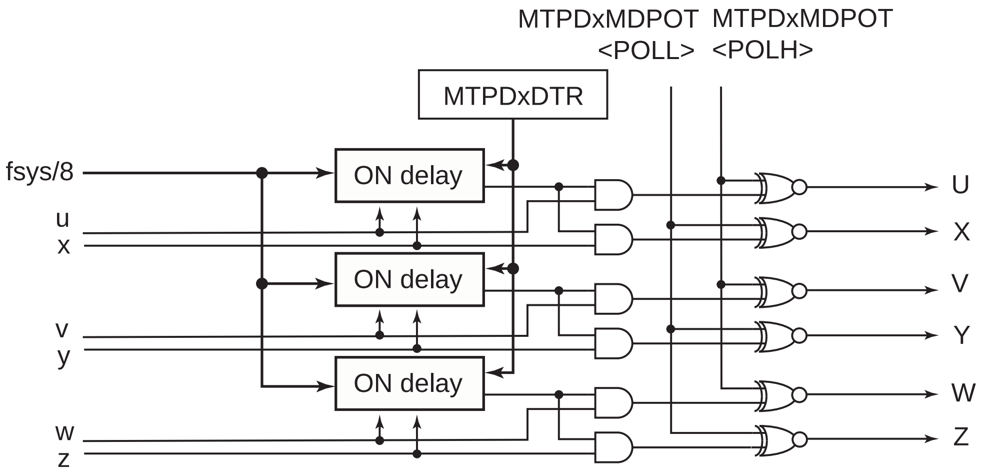


Figure 10-26 Schematic diagram of dead time circuit

The dead time circuit consists of the dead time circuit part and the output polarity switching part.

In each U-, V- and W-phase, this circuit prevent short circuit with delaying on-time using the dead time counter in case that upper phase and lower phase are reversed. A delay time is set to the dead time register (MTPDxDTR<DTR>) and can be set to 200ns @ fsys = 40MHz resolution with 8 bits.

The output polarity switching circuit can be set upper or lower signals to be high-active or low-active respectively using MTPDxMDPOT<POLH><POLL>.

10.9.5 Synchronous Trigger Generation Circuit

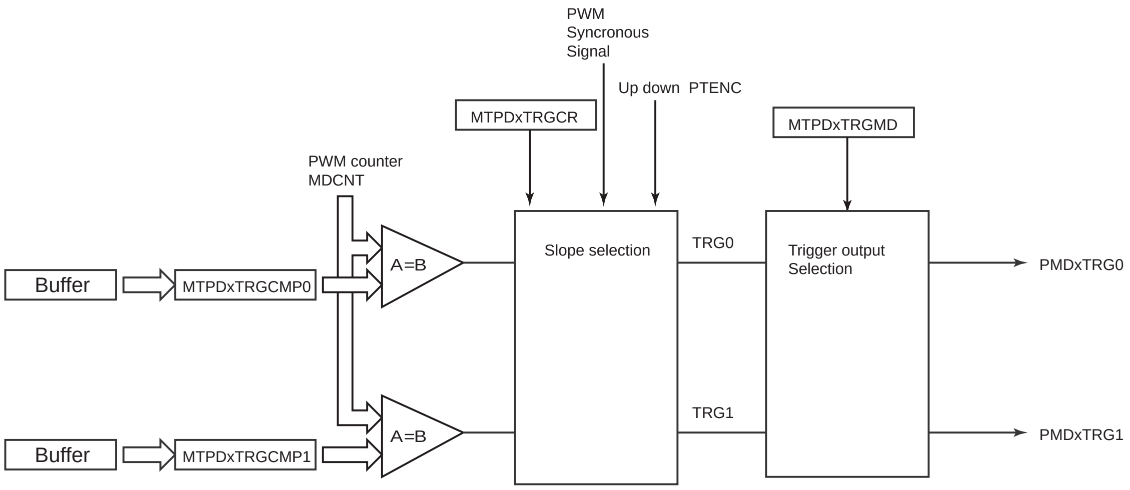


Figure 10-27 Schematic diagram of Synchronous Trigger Generation Circuit

The synchronous trigger generation circuit generates a trigger signal to sample AD converter synchronous with PWM. The operation is to generate a trigger signal PMDxTRG0 to 1 of AD converter when MTPDxMDCNT <MDCNT> matches MTPDxTRGCMP0<TRGCMP0>/MTPDxTRGCMP1<TRGCMP1>. The generation timing can be selectable among following match timings; a match in up-counting, a match in down-counting or a match in up-/down-counting. When the edge mode is selected, a generation timing is selected on a match in up-counting. When PWM output is disabled MTPDxMDEN<PWMEN> = "0"), a trigger is not output.

11. Serial Channel (SIO/UART)

11.1 Overview

This device has two modes for the serial channel, one is the synchronous communication mode (I/O interface mode), and the other is the asynchronous communication mode (UART mode).

Their features are described as follows.

- Transfer Clock
  - Generate the transfer clock by dividing the peripheral clock (φT0) frequency into 1/2, 1/8, 1/32, 1/128.
  - The prescaler output clock frequency can be divided by each of the numbers from 1 to 16.
  - The prescaler output frequency can be divided by each of the numbers from 1, N+m/16 (N=2 to 15, m=1 to 15), and 16. (only UART mode)
  - The system clock is usable. (only UART mode)
- Double buffer / FIFO

  The double buffer function and the FIFO buffers (total of transmit and receive) can be used up to 4bytes.
- I/O Interface mode
  - Transfer Mode : the half duplex (transmit / receive) and the full duplex
  - Clock : Output (fixed rising edge) / Input (selectable rising / falling edge)
  - A time interval can be set within a range where continuous transmission is performed.
- UART Mode
  - Data length : 7, 8, 9 bits
  - Add parity bit (to be against 9 bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{CTSx}}$ pin

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

11.2 Difference in the Specification of SIO / UART Modules

TMPM3U6FY/FW has 5 channels.

Each channel function is not depended. The pins and interrupts for each channel are assigned as follows.

Table 11-1 Differences for each channels of SIO / UART Modules

	Pin name			Interrupt		Timer for serial clock	DMA
	TXDx	RXDx	$\overline{\text{CTSx}}$ / SCLKx	Receive interrupt	Transmit interrupt		
channel 0	PE0	PE1	PE2	INTRX0	INTTX0	TB4OUT	support
channel 1	PA5	PA6	PA4	INTRX1	INTTX1	TB4OUT	support
channel 2	PD5	PD6	PD4	INTRX2	INTTX2	TB7OUT	support
channel 3	PF3	PF4	PF2	INTRX3	INTTX3	TB7OUT	support
channel 4	PC6	PC7	PC5	INTRX4	INTTX4	MTTB0OUT	support

11.3 Configuration

Figure 11-1 shows SIO / UART block diagram.

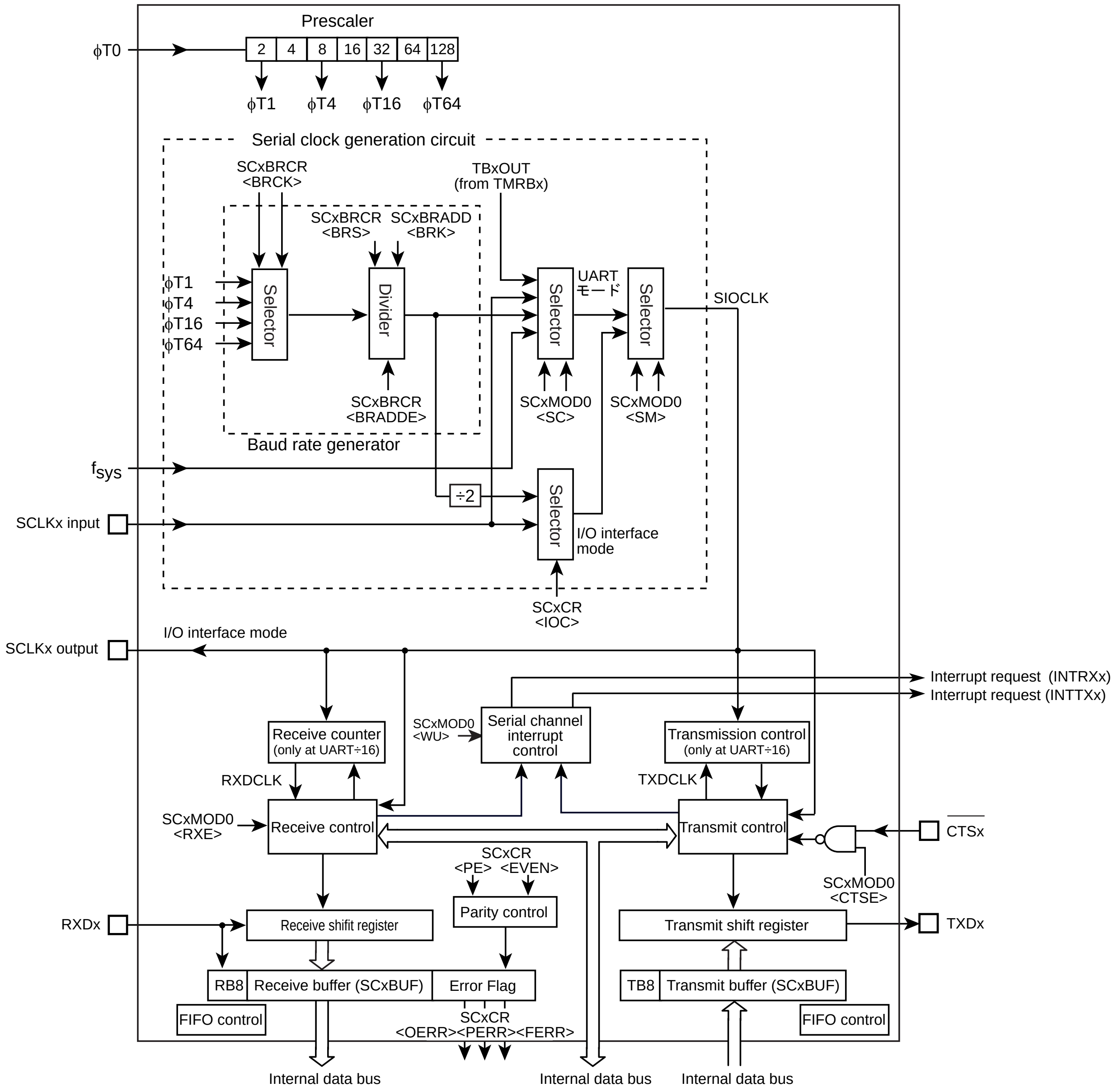


Figure 11-1 SIO / UART Block Diagram

11.4 Registers Description

11.4.1 Registers List in Each Channel

The below table shows registers and addresses for each register.

Channel x	Base Address
Channel0	0x4002_0080
Channel1	0x4002_00C0
Channel2	0x4002_0100
Channel3	0x4002_0140
Channel4	0x4002_0180

Register name (x=0 to 4)		Address (Base+)
Enable register	SCxEN	0x0000
Buffer register	SCxBUF	0x0004
Control register	SCxCR	0x0008
Mode control register 0	SCxMOD0	0x000C
Baud rate generator control register	SCxBRCR	0x0010
Baud rate generator control register 2	SCxBRADD	0x0014
Mode control register 1	SCxMOD1	0x0018
Mode control register 2	SCxMOD2	0x001C
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	-	-	-	-	-	-	-	SIOE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SIOE	R/W	SIO/UART operation 0: disable 1: Operation Specifies the SIO/UART operation. To use the SIO/UART, set <SIOE> = "1". When the operation is disabled, no clock is supplied to the other registers in the SIO/UART module. This can reduce the power consumption. If the SIO/UART operation is executed and then disabled, the settings will be maintained in each register except for SCxTFC<TIL>.

Note: In case that SCxEN<SIOE>="0" (Stop SIO/UART operation) or the operation mode is changed to IDLE mode with SCxMOD1<I2SC>="0" (Stop SIO/UART operation in IDLE mode), SCxTFC is initialized again.

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 / FIFO [read] RB : Receive buffer / 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RB8	R	Receive data bit 8 (for UART mode) 9th bit of the received data in the 9 bits UART mode.
6	EVEN	R/W	Parity (for UART mode) 0: Odd 1: Even Selects even or odd parity. "0" :odd parity, "1" : even parity The parity bit can be used only in the 7-bit or 8-bit UART mode.
5	PE	R/W	Adding parity (for UART mode) 0: Disabled 1: Enabled Controls enabling / disabling parity The parity bit can be used only in the 7-bit or 8-bit UART mode.
4	OERR	R	Overrun error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Underrun error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selecting input clock edge (for I/O Interface mode) Set to "0" in the clock output mode. 0:Data in the transmit buffer is sent to TXDx pin one bit at a time on the falling edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the rising edge of SCLKx. In this case, the state of 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 every one bit at a time on the falling edge of SCLKx. In this case, the state of SCLKx starts from low level.
0	IOC	R/W	Selecting clock (for I/O Interface mode) 0: Baud rate generator 1: SCLK pin input

Note:<OERR>, <PERR> and <FERR> are cleared to "0" when they are 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 mode) Writes the 9th bit of transmit data in the 9 bits UART mode.
6	CTSE	R/W	Handshake function control (For UART mode) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{\text{CTSx}}$ pin.
5	RXE	R/W	Receive control (Note1) (Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For UART mode) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. When it is set to be enabled, Interrupt occurs only when 9th bit = "1" at 9-bit in the UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For UART mode) 00: Timer TBxOUT Refer to Table 11-1. 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLKx 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 enable the <RXE> bit.

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 operation in the IDLE mode.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. And when FIFO is enabled, specify the configuration of FIFO. In the UART mode, specify the only configuration of FIFP.
4	TXE	R/W	Transmit control (Note1) (Note2) 0: Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface mode) 000: None 001: 1SCLK 010: 2SCLK 011: 4SCLK 100: 8SCLK 101: 16SCLK 110: 32SCLK 111: 64SCLK This parameter is valid only for the I/O interface mode when SCLK pin output is selected. In other modes, this function has no meaning. Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.
0	−	R/W	Write to "0".

Note 1: Specify all register first and then enable the <TXE> bit.

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

Note 3: In case that SCxEN<SIOE>="0" (Stop SIO/UART operation) or the operation mode is changed to IDLE mode with SCxMOD1<I2SC>="0" (Stop SIO/UART operation in IDLE mode), SCxTFC is initialized again.

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

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Overwriting "01" in place of "10" generates a software reset.
When a software reset is executed, the following bits are initialized and the transmit/receive circuit and the FIFO become initial state (Note1)(Note2).
<table>
<tr><th>Register</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td>&lt;RXE&gt;</td></tr>
<tr><td>SCxMOD1</td><td>&lt;TXE&gt;</td></tr>
<tr><td>SCxMOD2</td><td>&lt;TBEMP>, <RBFLL>, <TXRUN&gt;</td></tr>
<tr><td>SCxCR</td><td>&lt;OERR&gt;, <PERR>, <FERR&gt;</td></tr>
</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 to "0".
6	BRADDE	R/W	N + (16 − K)/16 divider function (Only for UART mode) 0: disabled 1: enabled This division function can only be used in the UART mode.
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00: φT1 01: φT4 10: φT16 11: φT64
3-0	BRS[3:0]	R/W	Division ratio "N" (note1)(note2) 0000: 16 0001: 1 0010: 2 : 1111: 15

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

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

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 mode) 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 in the UART mode)
<BRS>	Specify "N"	
<BRK>	No setting required	Specify "K" (Note2)
Division ratio	Divide by N	$N + \frac{(16 - K)}{16}$ division

Note 1: To use the "N + (16 - K)/16" division function, be sure to set <BRADDE> to "1" after setting the K value to <BRK>. The "N + (16 - K)/16" division function can only be used in the UART mode.

Note 2: Specifying "K = 0" is prohibited.

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	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	RFST	TFIE	RFIE	RXTXCNT	CNFG
After reset	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>-</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>7-5</td><td>-</td><td>R/W</td><td>Be sure to write "000".</td></tr>
<tr><td>4</td><td>RFST</td><td>R/W</td><td>Bytes used in Receive FIFO
0: Maximum
1: Same as FILL level of Receive FIFO
When Receive FIFO is enabled, the number of Receive FIFO bytes to be used is selected (Note1)
0: The maximum number of bytes of the FIFO configured (see also <CNFG>).
1: Same as the fill level for receive interrupt generation specified by SCxRFC <RIL[1:0]></td></tr>
<tr><td>3</td><td>TFIE</td><td>R/W</td><td>Specify transmit interrupt for transmit FIFO
0:Disabled
1:Enabled
When transmit FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.</td></tr>
<tr><td>2</td><td>RFIE</td><td>R/W</td><td>Specify receive interrupt for receive FIFO
0:Disabled
1:Enabled
When receive FIFO is enabled, receive interrupts are enabled or disabled by this parameter.</td></tr>
<tr><td>1</td><td>RXTXCNT</td><td>R/W</td><td>Automatic disable of SCxMOD0<RXE> / SCxMOD1<TXE>
0: None
1: Auto disabled
Controls automatic disabling of transmission and reception.
Setting "1" enables to operate as follows
<table>
<tr><td>Half duplex Receive</td><td>When the receive shift register, the receive buffer and the receive FIFO are filled up to the specified number of valid bytes, SCxMOD0<RXE> is automatically set to "0" to inhibit further reception.</td></tr>
<tr><td>Half duplex Transmit</td><td>When the transmit shift register , transmit buffers and the transmit FIFO are empty, SCxMOD1<TXE> is automatically set to "0" to inhibit further transmission.</td></tr>
<tr><td>Full duplex</td><td>When either of the above two conditions is satisfied, <TXE> and <RXE> are automatically set to "0" to inhibit further transmission and reception.</td></tr>
</table></td></tr>
<tr><td>0</td><td>CNFG</td><td>R/W</td><td>FIFO enable
0: Disabled
1: Enabled
Enables FIFO. (note2)
When <CNFG> is set to "1", FIFO is enabled. If FIFO is enabled, the SCxMOD1 <FDPX[1:0]> setting automatically configures FIFO as follows:
<table>
<tr><td>Half duplex Receive</td><td>Receive FIFO 4 bytes</td></tr>
<tr><td>Half duplex Transmit</td><td>Transmit FIFO 4 bytes</td></tr>
<tr><td>Full duplex</td><td>Receive FIFO 2 bytes + Transmit FIFO 2 bytes</td></tr>
</table></td></tr>
</table>

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

Note 2: The FIFO can not be used in 9bit 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	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFCS	RFIS	-	-	-	-	RIL	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RFCS	W	Receive FIFO clear (Note1) 1: Clear When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL[2:0]> is "000". And also the read pointer is initialized.
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]> Refer to 11.14.1.2 for details of the receive interrupt generation timing.
5-2	−	R	Read as "0".
1-0	RIL[1:0]	R/W	Receive FIFO fill level to generate receive interrupts

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

11.4.12 SCxTFC (Transmit FIFO Configuration Register) (Note2)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TFCS	W	Transmit FIFO clear (Note1) 0: Don’t care 1: Clear When SCxTFC<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL> is "000". And also the write pointer is initialized. Read as "0".
6	TFIS	R/W	Select interrupt generation condition 0: When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt <TIL [1:0]> 1: When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt <TIL [1:0]> Refer to 11.14.2.2 for details of the transmit interrupt generation timing.
5-2	−	R	Read as "0".
1-0	TIL[1:0]	R/W	Transmit FIFO fill level which transmit interrupt is occurred.

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

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

Note 2: In case that SCxEN<SIOE>="0" (Stop SIO/UART operation) or the operation mode is changed to IDLE mode with 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 (Note1) 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 1: <ROR> is cleared to "0" when receive data is read from the SCxBUF register.

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 register.

11.5 Operation in Each Mode

Table 11-3 shows the modes and data format.

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 (IO interface mode)	8 bit	LSB first / MSB first	-	-
Mode 1	Asynchronous communication mode (UART mode)	7 bit	LSB first	o	1 bit or 2 bits
Mode 2		8 bit		o	
Mode 3		9bit		×	

The 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 add a parity or STOP bits.

The mode 1, mode 2 and mode 3 are asynchronous modes and the transfer direction is fixed to the LSB first.

Parity bits can be added in the mode 1 and mode 2. The mode 3 has a wake-up function in which the master controller can start up slave controllers via the serial link (multi-controller system).

STOP bit in transmission can be selected from 1 bit and 2 bits. The STOP bit length in reception is fixed to a one bit.

11.6 Data Format

11.6.1 Data Format List

Figure 11-2 shows data format.

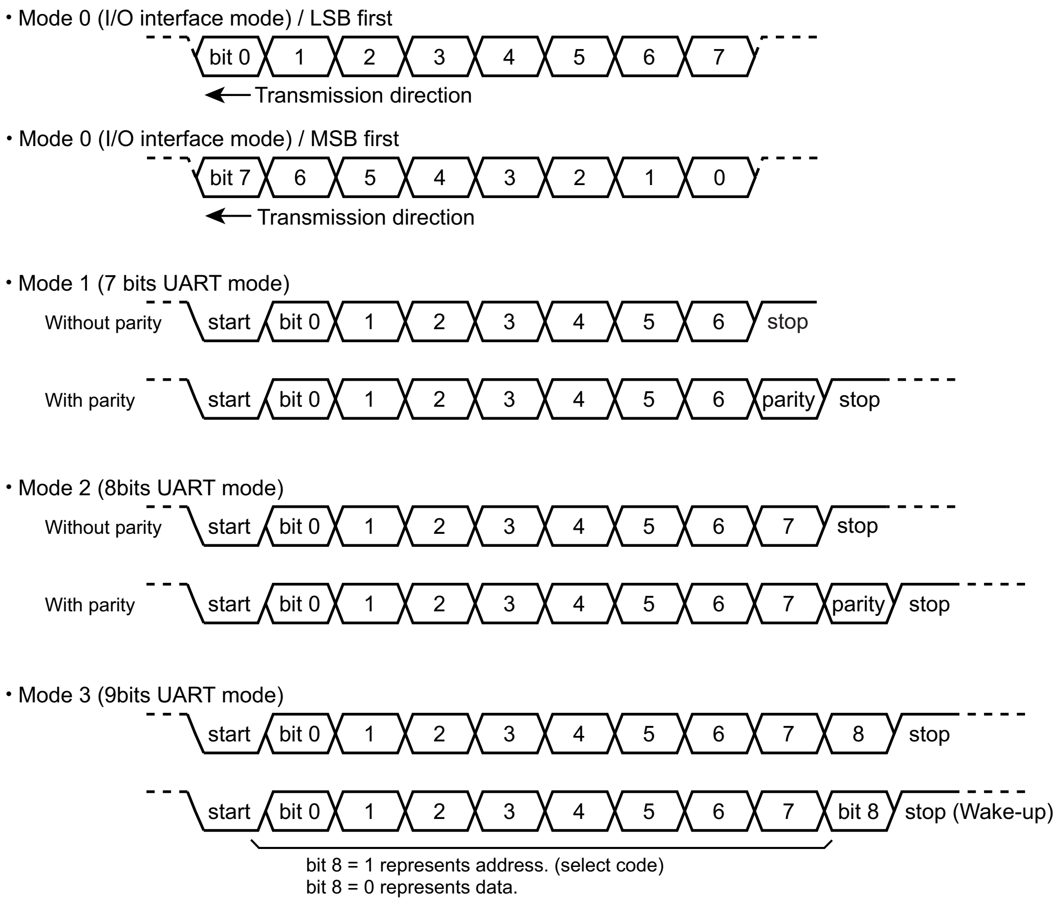


Figure 11-2 Data Format

11.6.2 Parity Control

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

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

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 SCxMOD0<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 Reception

If the received data is moved from the receive shift register to the receive buffer, a parity is generated.

In the 7-bit UART mode, the generated parity is compared with the parity stored in SCxBUF<RB[7]>, while in the 8-bit UART mode, it is compared with the one in SCxCR<RB8>.

If there is any difference, a parity error occurs and the 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

11.7.1 Prescaler

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

Use the CGSYSCR register in the clock / mode control block to select the input clock φT0 of the prescaler.

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

The below tables show the resolution of the input clock to the baud rate generator.

Table 11-4 Clock resolution to the Baud Rate Generator fc = 40 MHz

Peripheral clock selection	Clock gear value	Prescaler clock selection	Prescaler output clock resolution			
CGSYSCR <FPSEL[1:0]>	CGSYSCR <GEAR[2:0]>	CGSYSCR <PRCK[2:0]>	φT1	φT4	φT16	φT64
00 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^{1}$ (0.05 μs)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)
		001 (fperiph/2)	$fc/2^{2}$ (0.1 μs)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)
		010 (fperiph/4)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		011 (fperiph/8)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^{2}$ (0.1 μs)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)
		001 (fperiph/2)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		010 (fperiph/4)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		011 (fperiph/8)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		100 (fperiph/16)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		101 (fperiph/32)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^{3}$ (0.2 μs)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)
		001 (fperiph/2)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		010 (fperiph/4)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		011 (fperiph/8)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		100 (fperiph/16)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
		101 (fperiph/32)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4μs)	$fc/2^{14}$ (409.6 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^{4}$ (0.4 μs)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)
		001 (fperiph/2)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		010 (fperiph/4)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		011 (fperiph/8)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
		100 (fperiph/16)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4μs)	$fc/2^{14}$ (409.6 μs)
		101 (fperiph/32)	$fc/2^{9}$ (12.8μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{15}$ (819.2 μs)
	111 (fc/16)	000 (fperiph/1)	$fc/2^{5}$ (0.8 μs)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)
		001 (fperiph/2)	$fc/2^{6}$ (1.6 μs)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4 μs)
		010 (fperiph/4)	$fc/2^{7}$ (3.2 μs)	$fc/2^{9}$ (12.8 μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)
		011 (fperiph/8)	$fc/2^{8}$ (6.4 μs)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4μs)	$fc/2^{14}$ (409.6 μs)
		100 (fperiph/16)	$fc/2^{9}$ (12.8μs)	$fc/2^{11}$ (51.2 μs)	$fc/2^{13}$ (204.8 μs)	$fc/2^{15}$ (819.2 μs)
		101 (fperiph/32)	$fc/2^{10}$ (25.6 μs)	$fc/2^{12}$ (102.4μs)	$fc/2^{14}$ (409.6 μs)	$fc/2^{16}$ (1638.4 μs)

Table 11-4 Clock resolution to the Baud Rate Generator fc = 40 MHz

Peripheral clock selection CGSYSCR <FPSEL[1:0]>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
01 (fc)	000 (fc)	000 (fperiph/1)	fc/2^{1} (0.05 μs)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	fc/2^{2} (0.1 μs)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	fc/2^{2} (0.1 μs)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	fc/2^{3} (0.2 μs)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	–	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	fc/2^{4} (0.4 μs)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)
		001 (fperiph/2)	–	–	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)
		010 (fperiph/4)	–	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)
		011 (fperiph/8)	–	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)
		100 (fperiph/16)	fc/2^{5} (0.8 μs)	fc/2^{7} (3.2 μs)	fc/2^{9} (12.8 μs)	fc/2^{11} (51.2 μs)
		101 (fperiph/32)	fc/2^{6} (1.6 μs)	fc/2^{8} (6.4 μs)	fc/2^{10} (25.6 μs)	fc/2^{12} (102.4 μs)

Note 1: The prescaler output clock φTn must be selected so that the relationship "φTn ≤ fsys/2" is satisfied (so that φTn is slower than fsys).

Note 2: Do not change the clock gear while SIO/UART is operating.

Note 3: The "−" indicates that the setting is prohibited in the above table.

11.7.2 Serial Clock Generation Circuit

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

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 2, 8, 32 and 128.

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

(2) Baud Rate Generator output clock

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

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

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

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divided by N SCxBRCR<BRS>	Divided by K SCxBRADD<BRK>
I/O interface mode	Divide by N	1 to 16 (Note)	-
UART mode	Divide by N	1 to 16	-
	N + (16-K)/16 division	2 to 15	1 to 15

Note: 1/N (N=1)frequency division ratio can be used only when a double buffer is enabled.

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 input clock in I/O interface mode is selected by setting SCxCR.

The clock in UART mode is selected by setting SCxMOD0<SC>.

(1) Transfer Clock in I/O interface mode

Table 11-5 shows clock selection in I/O interface mode.

Table 11-5 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	SCLKx output	Set to "0" (Fixed to the rising edge)	Divided by 2 of the baud rate generator output
	SCLKx input	Rising edge	SCLKx input rising edge
		Falling edge	SCLKx input falling edge

To get the highest baud rate, the baud rate generator must be set as below.

Note: When deciding clock settings, make sure that AC electrical character is satisfied.

- Clock / mode control block settings
  - fc = 40MHz
  - fgear = 40MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0=40MHz (CGSYSCR<PRCK[2:0]>="000" : 1 division ratio)
- SIO/UART settings (if double buffer is used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected) = 20MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0001":1 division ratio) = 20MHz

  1 division ratio can be selected if double buffer is used. In this case, baud rate is 10Mbps because 20MHz is divided by 2.
- SIO/UART settings (if double buffer is not used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00":φT1 selected) = 20MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0010" : 2 division ratio) = 10MHz

  2 division ratio is the highest if double buffer is not used. In this case, baud rate is 5Mbps because 10MHz is divided by 2.

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

- If double buffer is used

  - SCLK cycle > 6/fsys

  The highest baud rate is less than 40 ÷ 6 = 6.66 Mbps.
- If double buffer is not used

  - SCLK cycle > 8/fsys

  The highest baud rate is less than 40 ÷ 8 = 5 Mbps.

(2) Transfer clock in the UART mode

Table 11-6 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 11-6 Clock selection in UART Mode

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

The examples of baud rate in each clock settings.

- If baud rate generator is used.
  - fc = 40MHz
  - fgear = 40MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 40MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
  - Clock = φT1 = 20MHz (SCxBRCR<BRCK[1:0]> = "00" : φT1 selected)

  The highest baud rate is 1.25MHz because 20MHz is divided by 16.

  Table 11-7 shows examples of baud rate when the baud rate generator is used with the following clock settings.

  - fc = 9.8304MHz
  - fgear = 9.8304MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - φT0 = 4.9152MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)

Table 11-7 Example of UART Mode Baud Rate (Using the Baud Rate Generator)

fc [MHz]	Division ratio N (SCxBRCR<BRS[3:0]>)	φT1 (fc/4)	φT4 (fc/16)	φT16 (fc/64)	φT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	16	9.600	2.400	0.600	0.150

Unit : kbps

- If the SCLKx input is used

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

  - SCLK cycle > 2/fsys

  The highest baud rate must be less than 40 ÷ 2 ÷ 16 = 1.25 Mbps.

- If fsys is used

  Since the highest value of fsys is 40MHz, the highest baud rate is 40 ÷ 16 = 2.5Mbps.

- If timer output is used

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

  Baud rate can be obtained by using the following formula.

Baud rate calculation

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

In the case the timer prescaler clock fT1(2 division ratio) is selected

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

Table 11-8 shows the examples of baud rates when the timer output is used with the following clock settings.

- fc = 32MHz / 9.8304MHz / 8MHz
- fgear = 32MHz / 9.8304MHz / 8MHz (CGSYSCR<GEAR[2:0]> = "000" :fc selected)
- φT0 = 16MHz / 4.9152MHz / 4MHz (CGSYSCR<PRCK[2:0]> = "001" :2 division)
- Timer count clock= 4MHz / 1.2287MHz / 1MHz (TBxMOD<TBCLK[1:0]> = "01" :φT1 selected)

Table 11-8 Example of UART Mode Baud Rate (Using the Timer Output)

TBxRG1 setting	fc		
	32MHz	9.8304MHz	8MHz
0x0001	250	76.8	62.5
0x0002	125	38.4	31.25
0x0003	-	25.6	-
0x0004	62.5	19.2	15.625
0x0005	50	15.36	12.5
0x0006	-	12.8	-
0x0008	31.25	9.6	-
0x000A	25	7.68	6.25
0x0010	15.625	4.8	-
0x0014	12.5	3.84	3.125

Unit : kbps

11.8 Transmit / Receive Buffer and FIFO

11.8.1 Configuration

Figure 11-3 shows the configuration of transmit buffer, receive buffer and FIFO.

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

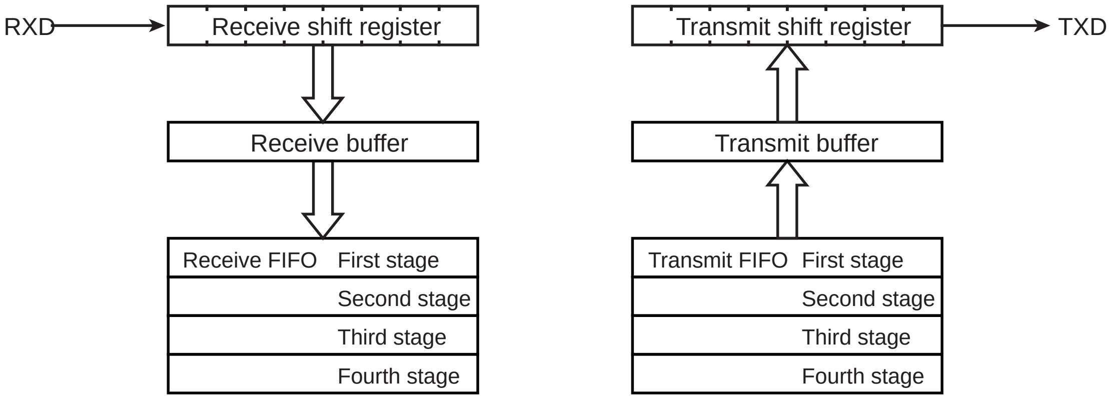


Figure 11-3 The Configuration of Buffer and FIFO

11.8.2 Transmit / Receive Buffer

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

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

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

Table 11-9 shows correlation between modes and buffers.

Table 11-9 Mode and buffer Composition

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

11.8.3 FIFO

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

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

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

Table 11-10 shows correction between modes and FIFO.

Table 11-10 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

11.9 Status Flag

The SCxMOD2 register has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1". When reading the receive bufferes, this bit is cleared to "0".

<TBEMP> shows that the transmit buffer is empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1". When data is set to the transmit buffers, the bit is cleared to "0".

11.10 Error Flag

Three error flags are provided in the SCxCR register. The meaning of the flags is changed depending on the modes. The table below shows the meaning in each mode.

These flags are cleared to "0" after reading the SCxCR register.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART mode	Overrun error	Parity error	Framing error
I/O interface mode (SCLKx input)	Overrun error	Underrun error (When using double buffer or FIFO)	Fixed to "0"
		Fixed to "0" (When a double buffer and FIFO unused)	
I/O interface mode (SCLKx output)	Undefined	Undefined	Fixed to "0"

11.10.1 OERR Flag

In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame of receive data before the receive buffer has been read.

If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied).

In the I/O interface mode with SCLK output mode, the SCLKx output stops upon setting the flag.

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

11.10.2 PERR Flag

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

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

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

In the 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 from the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

11.10.3 FERR Flag

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

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

11.11 Receive

11.11.1 Receive Counter

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

In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

11.11.2 Receive Control Unit

11.11.2.1 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 the shift clock outputted to the SCLKx pin.

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

11.11.2.2 UART Mode

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

11.11.3 Receive Operation

11.11.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is cleared to "0" by reading the receive buffer. When the double buffer is disabled, the receive buffer full flag has no meaning.

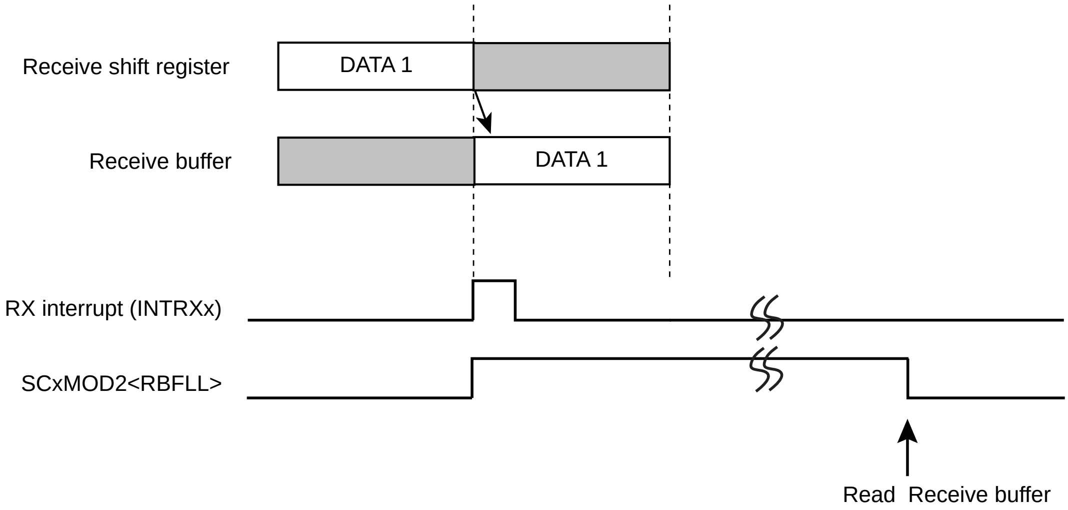


Figure 11-4 Receive Buffer Operation

11.11.3.2 Receive FIFO Operation

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

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

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

SCxMOD1[6:5] =01	:Transfer mode is set to half duplex mode
SCxFCNF[4:0] = 10111	:Automatically inhibits continuous reception after reaching the fill level.
	: The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxRFC[1:0] = 00	:The fill level of FIFO in which generated receive interrupt is set to 4-bytes
SCxRFC[7:6] = 11	:Clears receive FIFO and sets the condition of interrupt generation.

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

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

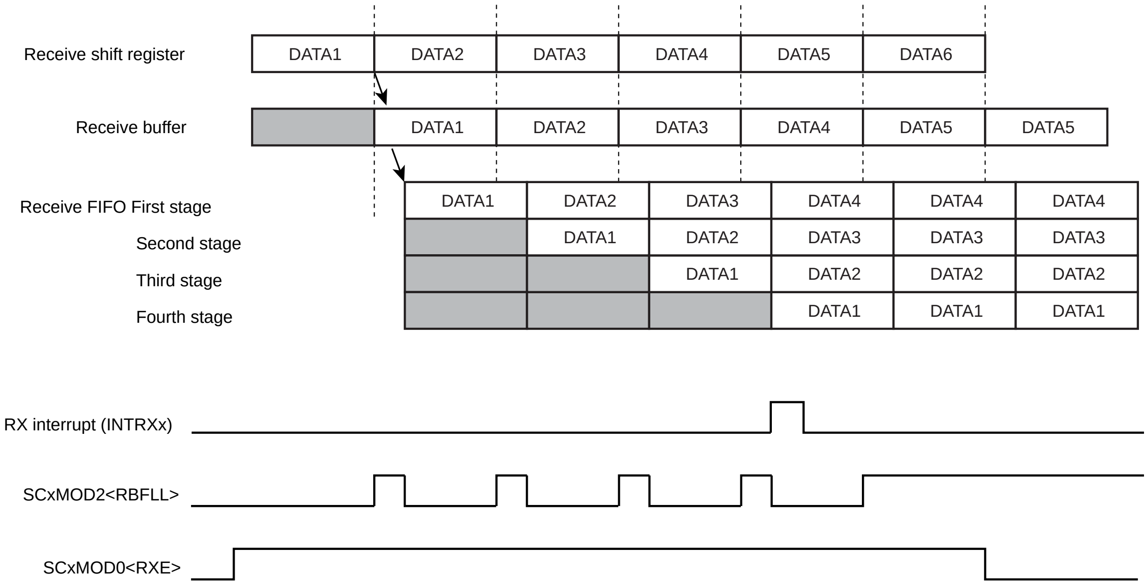


Figure 11-5 Receive FIFO Operation

11.11.3.3 I/O interface mode with SCLKx output

In the I/O interface mode and SCLKx output setting, SCLKx 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 SCLKx output stop and re-output depends on receive buffer and FIFO.

(1) Case of single buffer

Stop SCLKx output after receiving a data. In this mode, I/O interface mode can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, SCLKx output is restarted.

(2) Case of double buffer

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

When the data is read, SCLKx output is restarted.

(3) Case of FIFO

Stop SCLKx 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 SCLKx output restarts.

And if SCxFCNF<RXTXCNT>is set to "1", SCLKx stops and receive operation stops with clear-ing SCxMOD0<RXE> bit, too.

11.11.3.4 Read Received Data

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

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. The next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

When the receive FIFO is enabled, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in receive FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

11.11.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1". In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

11.11.3.6 Overrun Error

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

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

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

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

11.12 Transmission

11.12.1 Transmission Counter

The transmit counter is a 4-bit binary counter and is counted by SIOCLK as in the case of the receive counter.

In UART mode, it generates a transmit clock (TXDCLK) on every 16th clock pulse.

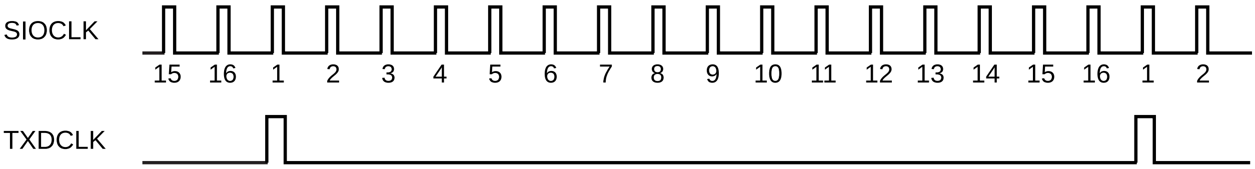


Figure 11-6 Generation of Transmission Clock in UART Mode

11.12.2 Transmission Control

11.12.2.1 I/O interface Mode

In the SCLK output mode with SCxCR<IOC> set to "0", each bit of data in the transmit buffer is outputted to the TXDx pin on the falling edge of the shift clock outputted from the 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 input signal according to the SCxCR<SCLKS> setting.

11.12.2.2 UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

11.12.3 Transmit Operation

11.12.3.1 Operation of Transmission Buffer

If double buffering is disabled, the CPU writes data only to Transmit shift Register and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled (including the case the transmit FIFO is enabled), data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

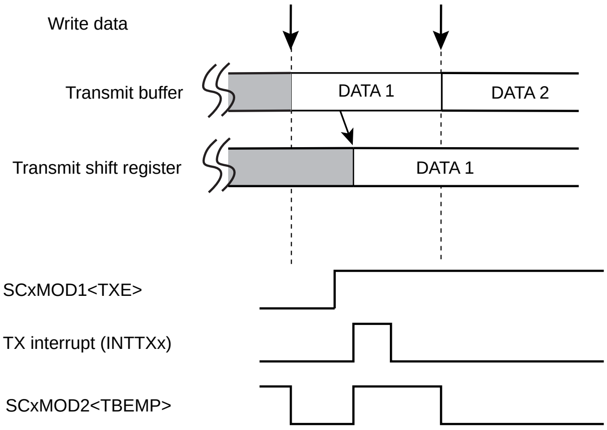


Figure 11-7 Operation of Transmission Buffer (Double buffer is enabled)

11.12.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

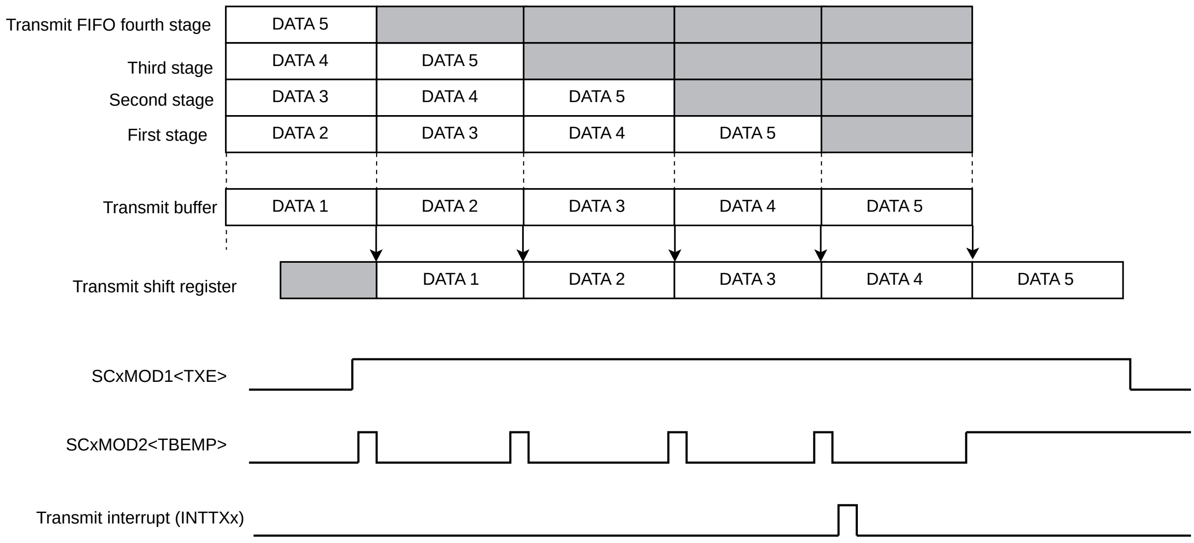
Note: **To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO/UART transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").**

Settings and operations to transmit 4-byte data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1[6:5] =10	:Transfer mode is set to half duplex.
SCxFCNF[4:0] = 11011	:Transmission is automatically disabled if FIFO becomes empty. :The number of bytes to be used in the receive FIFO is the same as the interrupt :generation fill level.
SCxTFC[1:0] = 00	:Sets the interrupt generation fill level to "0".
SCxTFC[7:6] = 11	:Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF[0] = 1	:Enable FIFO

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

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts writing transmit data.



11.12.3.3 I/O interface Mode/Transmission by SCLK Output

If SCLK is set to generate clock in the I/O interface mode, the SCLKx output automatically stops when all data transmission is completed and underrun error will not occur.

The timing of suspension and resume of SCLKx output is different depending on the buffer and FIFO us-age.

(1) Single Buffer

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

(2) Double Buffer

The SCLKx output stops upon completion of data transmission of the transmit shift register and the transmit buffer. The SCLKx 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 SCLKx output stops. The next data is written, SCLKx output resumes.

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

11.12.3.4 Underrun Error

If the transmit FIFO is disabled in the I/O interface mode with SCLK input mode and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from trans-mit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

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

Note: **Before switching the I/O interface mode SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

11.13 Handshake Function

The function of the handshake is to enable frame-by-frame data transmission by using the CTS (Clear to send) pin and to prevent overrun errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTSx}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until $\overline{\text{CTSx}}$ pin returns to the "Low" level. However in this case, the INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note 1: **If the $\overline{\text{CTSx}}$ signal is set to "High" during transmission, the next data transmission is suspended after the current transmission is completed (Point "a" in Figure 11-9).**

Note 2: **Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTSx}}$ is set to "Low" (Point "b" in Figure 11-9).**

Although no $\overline{\text{RTS}}$ pin is provided, a handshake control function can easily implemented by assigning one bit of the port for the $\overline{\text{RTS}}$ function. By setting the port to "High" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

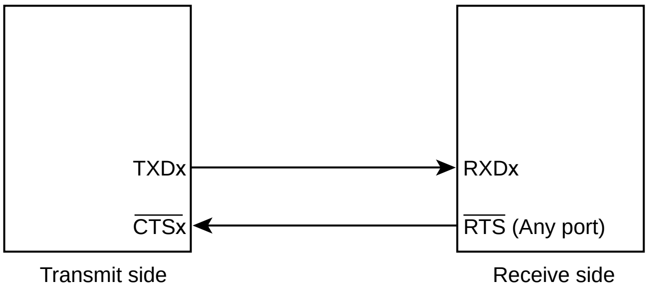


Figure 11-8 Handshake Function

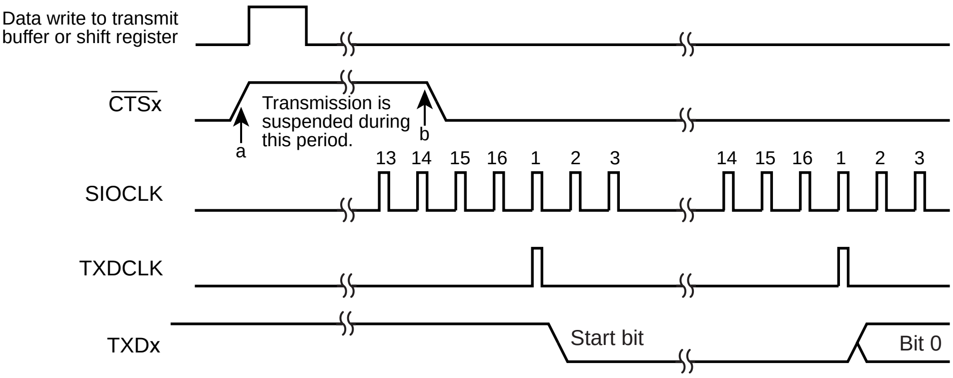

Data write to transmit buffer or shift register
$\overline{\text{CTSx}}$
Transmission is suspended during this period.
a
b
13 14 15 16 1 2 3
14 15 16 1 2 3
SIOCLK
TXDCLK
TXDx
Start bit
Bit 0

Figure 11-9 $\overline{\text{CTSx}}$ Signal timing

11.14 Interrupt / Error Generation Timing

11.14.1 RX Interrupt

Figure 11-10 shows the data flow of receive operation and the route of read.

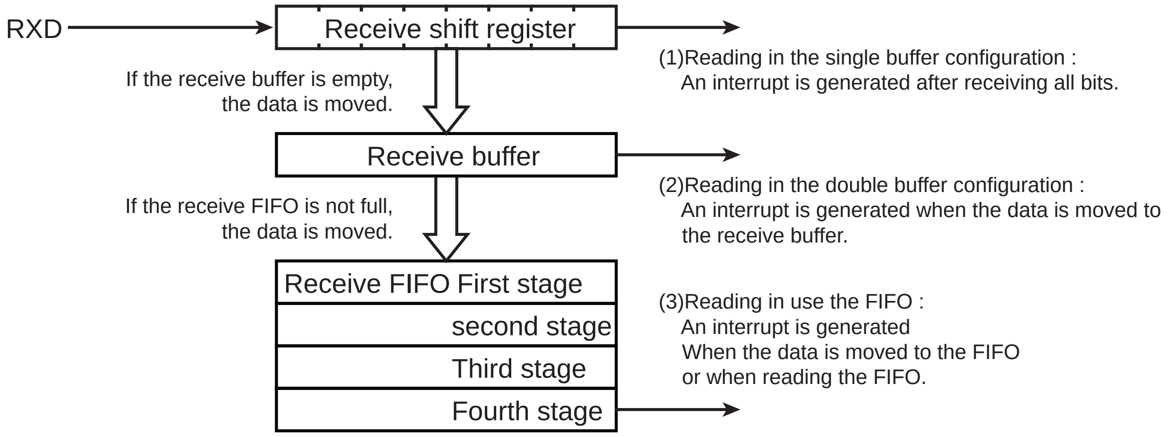

RXD
Receive shift register
If the receive buffer is empty, the data is moved.
Receive buffer
If the receive FIFO is not full, the data is moved.
Receive FIFO First stage
second stage
Third stage
Fourth stage
(1)Reading in the single buffer configuration : An interrupt is generated after receiving all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

Figure 11-10 Receive Buffer / FIFO Configuration Diagram

11.14.1.1 Single Buffer / Double Buffer

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

Table 11-11 Transmit interrupt condition with single buffer and double buffers

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

Note: Interrupts are not generated when an overrun error occurs.

11.14.1.2 FIFO

When the FIFO is used, a receive interrupt occurs depending on the timing described in Table 11-12 and the condition specified with SCxRFC<RFIS>.

Table 11-12 Receive Interrupt Conditions in use of FIFO

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

11.14.2 Transmit interrupt

Figure 11-11 shows the data flow of transmit operation and the route of write.

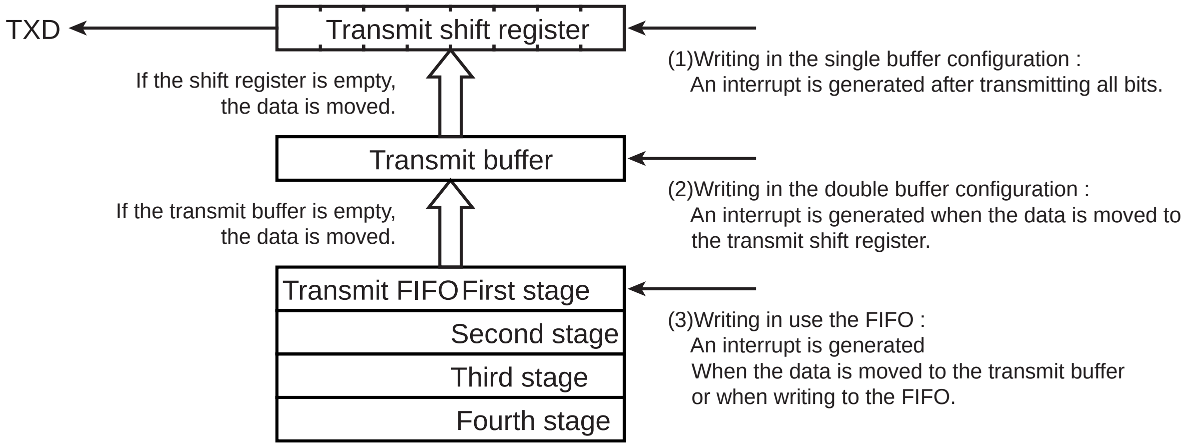

TXD
Transmit shift register
If the shift register is empty, the data is moved.
Transmit buffer
If the transmit buffer is empty, the data is moved.
Transmit FIFO First stage
Second stage
Third stage
Fourth stage
(1)Writing in the single buffer configuration : An interrupt is generated after transmitting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

Figure 11-11 Transmit Buffer / FIFO Configuration Diagram

11.14.2.1 Single Buffer / Double Buffer

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

Table 11-13 Receive interrupt condition with single buffer and double buffers

Buffer Configuration	UART modes	I/O interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the rising / falling edge of the last SCLKx. (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 with SCxMOD1<TXE>=1, transmit interrupt is generated because a data written to transfer buffer is moved from transmit buffer to transmit shift register.	

11.14.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 11-14 and the condition specified with SCxTFC<TFIS>

Table 11-14 Transmit Interrupt conditions in use of FIFO

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

11.14.3 Error Generation

11.14.3.1 UART Mode

Modes	9 bits	7 bits 8 bits 7 bits + parity 8bits + parity
Framing error Overrun error	Around the center of stop bit	
Parity Error	−	Determination: Around center of parity bit Flag change: Around the center of stop bit

11.14.3.2 I/O Interface Mode

Overrun error	Immediately after the rising / falling edge of the last SCLKx (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Underrun error	Immediately after the rising or falling edge of the next SCLKx (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.**

11.15 DMA Transfer

DMA transfer can be started at the timing of interrupt request.

TMPM3U6FY/FW can be started at the half duplex mode (Transmit and Receive interrupt request). But DMA transfer can not be started by the full duplex mode. In this case, the interrupt request is not specified as the source of DMA transfer request. And DMA transfer is started not depend on SCxFCNF<TFIE><RFIE>.

Note 1: In case using DMA transfer by transmit or receive interrupt request, enabled DMA and set transmit and receive registers after generating software reset by SCxMOD2<SWRST>.

Note 2: When the DMA transfer is used, the FIFO cannot be used.

Note 3: If transmission is performed using the DMA transfer with double-buffering, write transmission data to the buffer, and then start up the DMA.

11.16 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01". As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR<OERR><PERR><FERR> are initialized. And the receive circuit, the transmit circuit and the FIFO become initial state. Other states are held.

11.17 Operation in Each Mode

11.17.1 I/O Interface Mode

The I/O interface mode can be selected by setting the mode control register (SCxMOD0<SM[1:0]>) to "00".

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

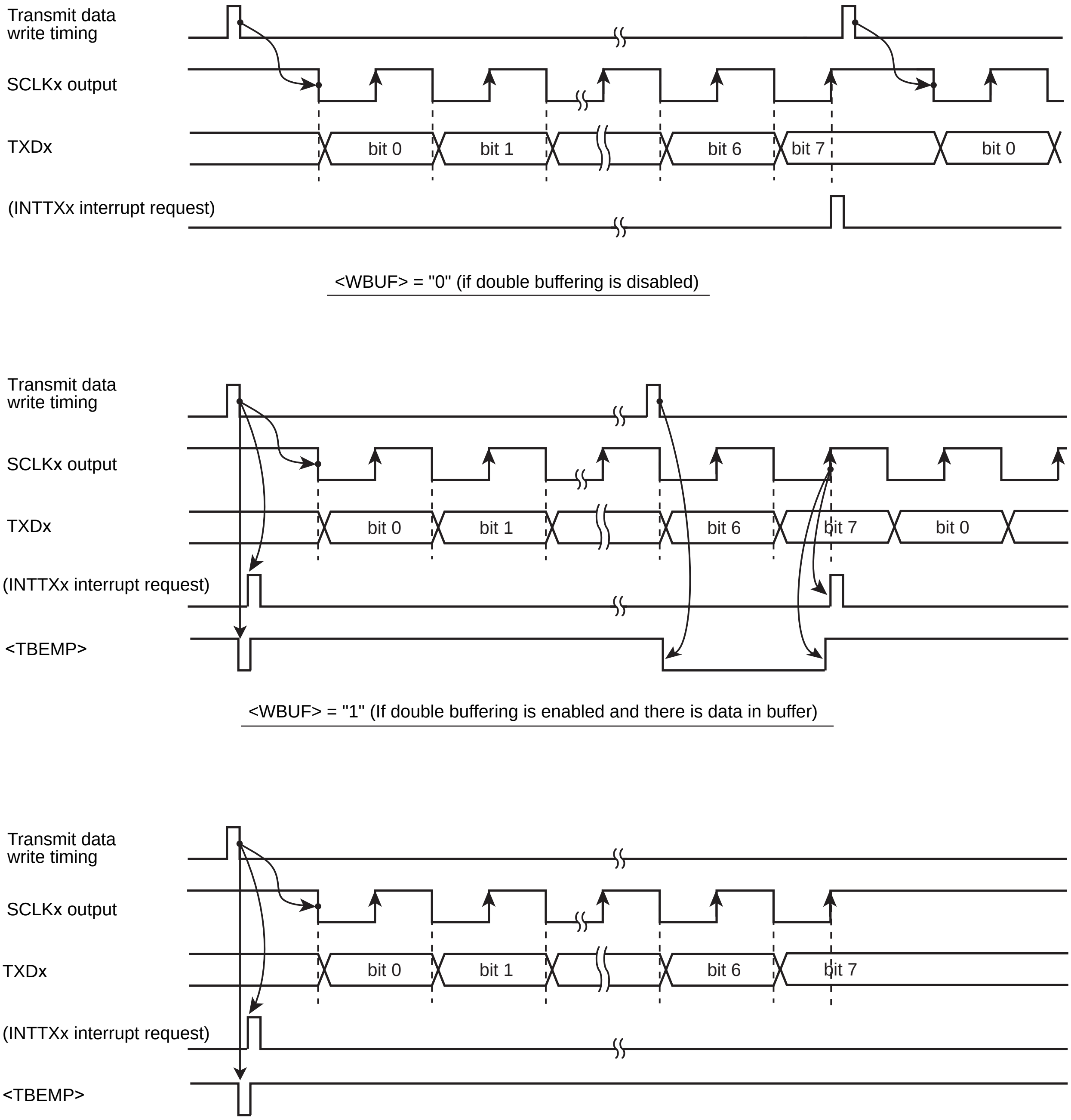
The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

11.17.1.1 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 SCLKx output stops.


Transmit data write timing
SCLKx output
TXDx
bit 0
bit 1
bit 6
bit 7
bit 0
(INTTXx interrupt request)
Transmit data write timing
SCLKx output
TXDx
bit 0
bit 1
bit 6
bit 7
bit 0
(INTTXx interrupt request)
<TBEMP>
Transmit data write timing
SCLKx output
TXDx
bit 0
bit 1
bit 6
bit 7
(INTTXx interrupt request)
<TBEMP>

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

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

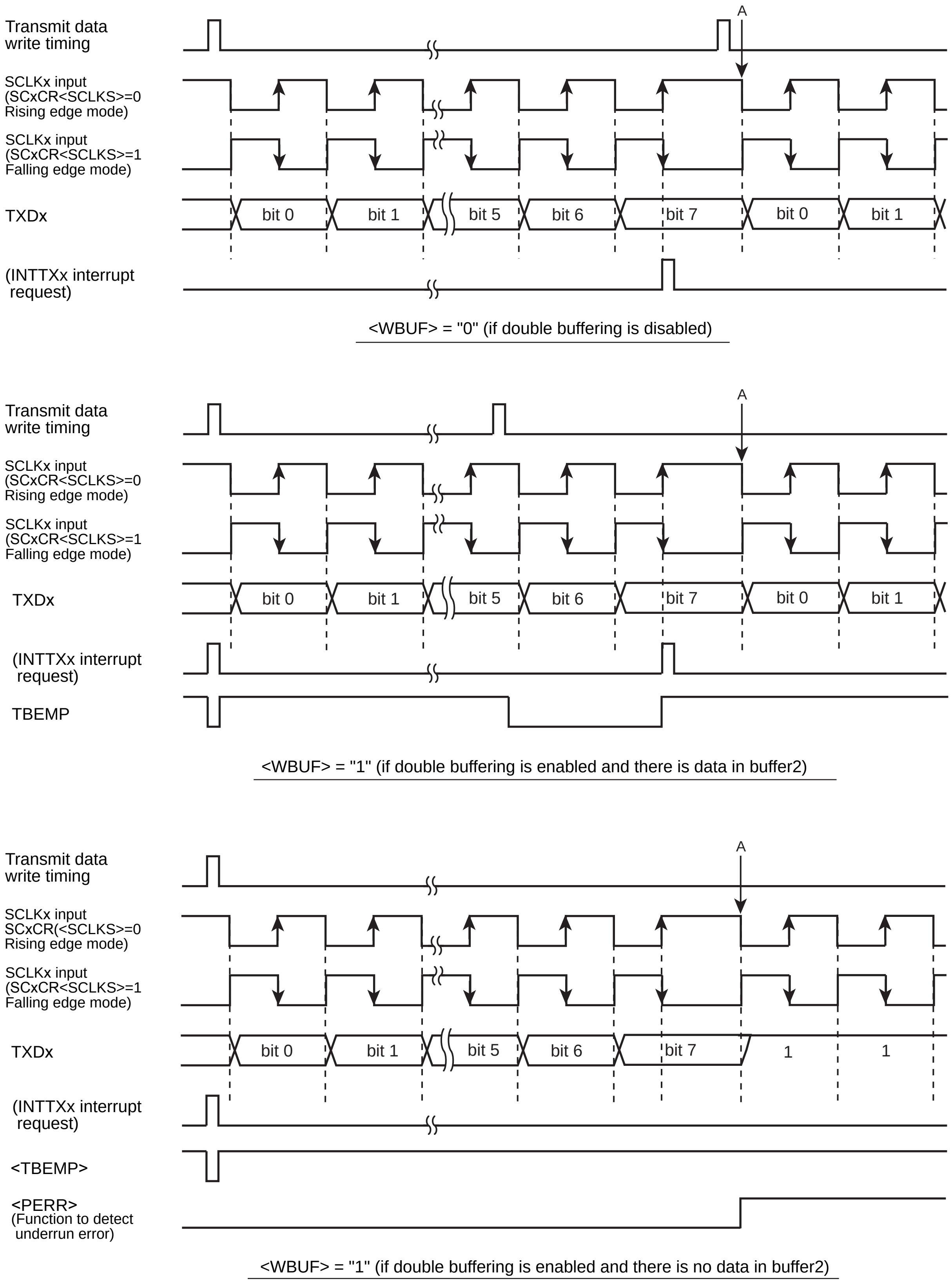
<WBUF> = "1" (if double buffering is enabled and there is no data in buffer)

Figure 11-12 Transmit Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If double buffering is disabled (SCxMOD2<WBUF> = "0")
  If the SCLKx 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 11-13.
- If double buffer is enabled (SCxMOD2<WBUF> = "1")
  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the SCLKx 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 SCLKx input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.


Transmit data write timing
A
SCLKx input (SCxCR<SCLKS>=0 Rising edge mode)
SCLKx input (SCxCR<SCLKS>=1 Falling edge mode)
TXDx
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
Transmit data write timing
A
SCLKx input (SCxCR<SCLKS>=0 Rising edge mode)
SCLKx input (SCxCR<SCLKS>=1 Falling edge mode)
TXDx
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
(INTTXx interrupt request)
TBEMP
Transmit data write timing
A
SCLKx input SCxCR(<SCLKS>=0 Rising edge mode)
SCLKx input (SCxCR<SCLKS>=1 Falling edge mode)
TXDx
bit 0
bit 1
bit 5
bit 6
bit 7
1
1
(INTTXx interrupt request)
<TBEMP>
<PERR> (Function to detect underrun error)

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

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

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

Figure 11-13 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

11.17.1.2 Receive

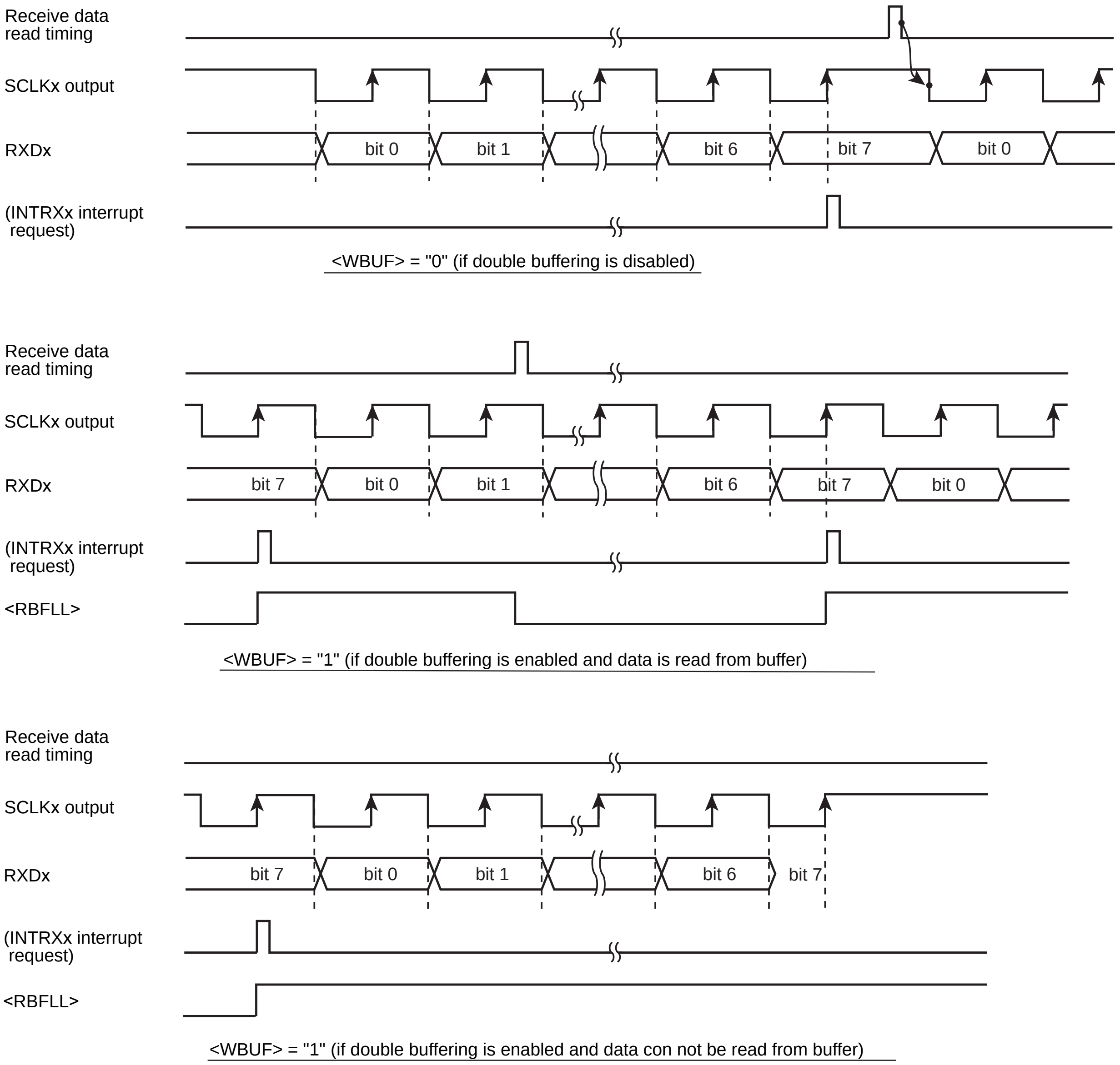
(1) SCLK Output Mode

The SCLKx 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 receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLKx output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.



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

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

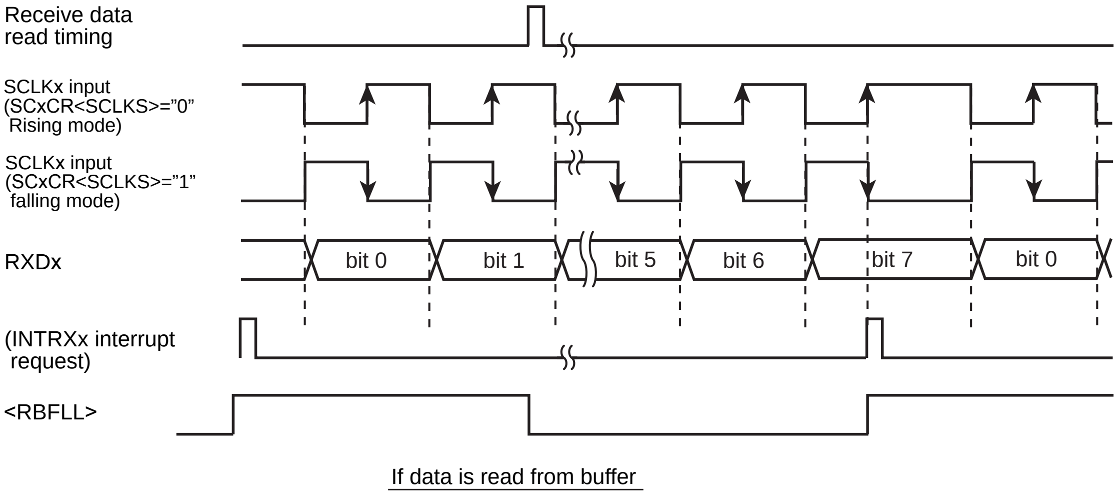
<WBUF> = "1" (if double buffering is enabled and data con not be read from buffer)

Figure 11-14 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to the receive buffer from the shift register, and the receive buffer can receive the next frame successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.


Receive data read timing
SCLKx input (SCxCR<SCLKS>="0" Rising mode)
SCLKx input (SCxCR<SCLKS>="1" falling mode)
RXDx
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
(INTRXx interrupt request)
<RBFLL>
If data is read from buffer

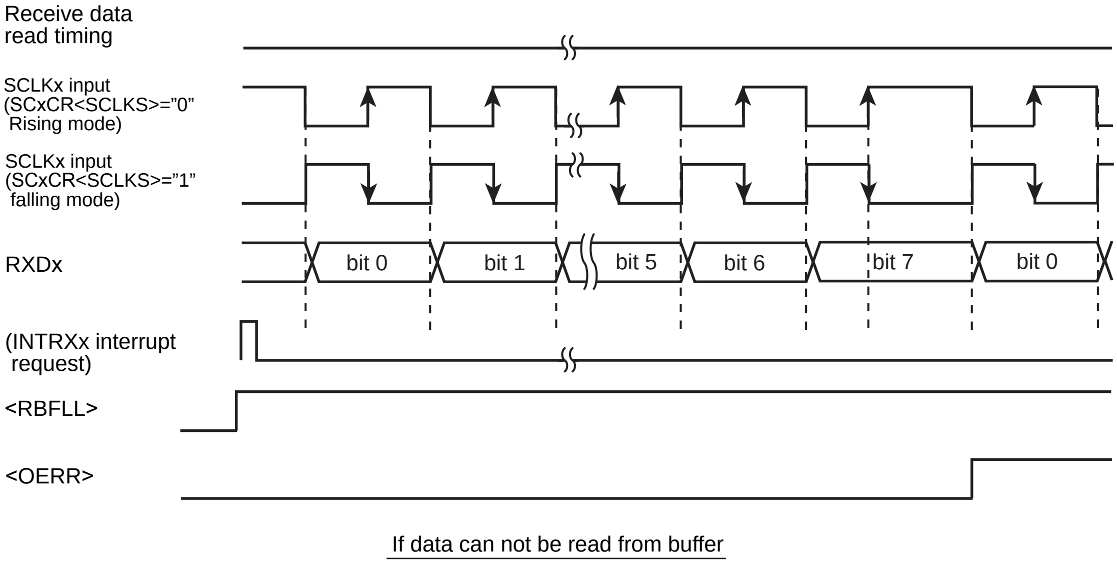

Receive data read timing
SCLKx input (SCxCR<SCLKS>="0" Rising mode)
SCLKx input (SCxCR<SCLKS>="1" falling mode)
RXDx
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
(INTRXx interrupt request)
<RBFLL>
<OERR>
If data can not be read from buffer

Figure 11-15 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

11.17.1.3 Transmit and Receive (Full duplex)

(1) SCLK Output Mode

- If SCxMOD2<WBUF> is set to "0" and the double buffers are disabled

  SCLKx is outputted when the CPU writes data to the transmit buffer.

  Subsequently, 8 bits of data are shifted into receive shift refister and the INTRXx receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are outputted from the TXDx pin, the INTTXx transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLKx output stops.

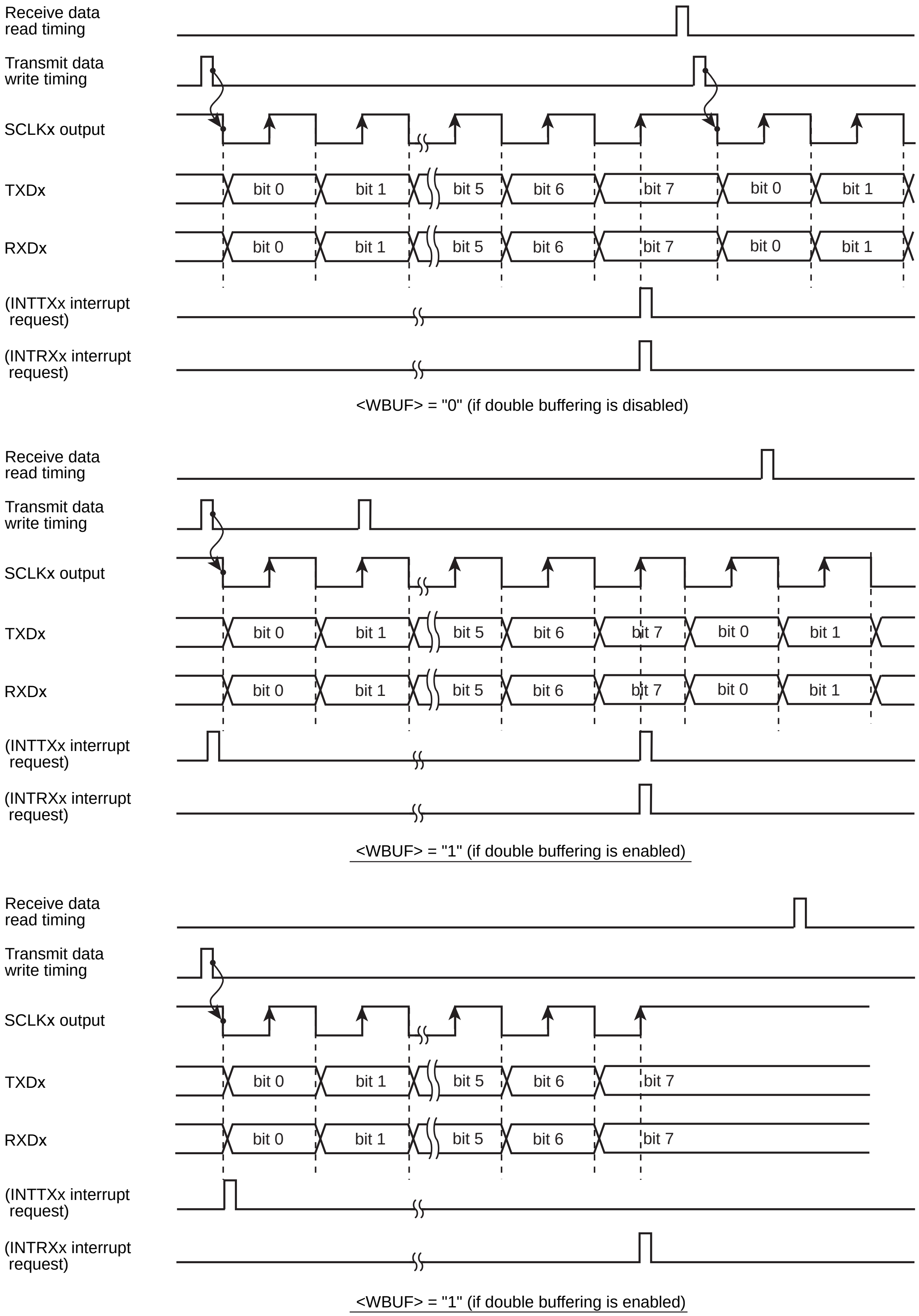
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If SCxMOD2<WBUF> is set to "1" and the double buffers are enabled

  SCLKx 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 SCLKx output is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLKx output is resumed and the next round of data transmission and reception is started.



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

<WBUF> = "1" (if double buffering is enabled)

<WBUF> = "1" (if double buffering is enabled)

Figure 11-16 Transmit / Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If SCxMOD2<WBUF> is set to "0" and the transmit double buffer is disabled
  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  8-bit data written in the transmit buffer is outputted from the TXDx pin and 8 bit of data is shifted into the receive buffer when the SCLKx input becomes active.The INTTXx interrupt is generated upon completion of data transmission. The INTRXx interrupt is gener-ated when the data is moved from receive shift register to receive buffer after completion of data reception.

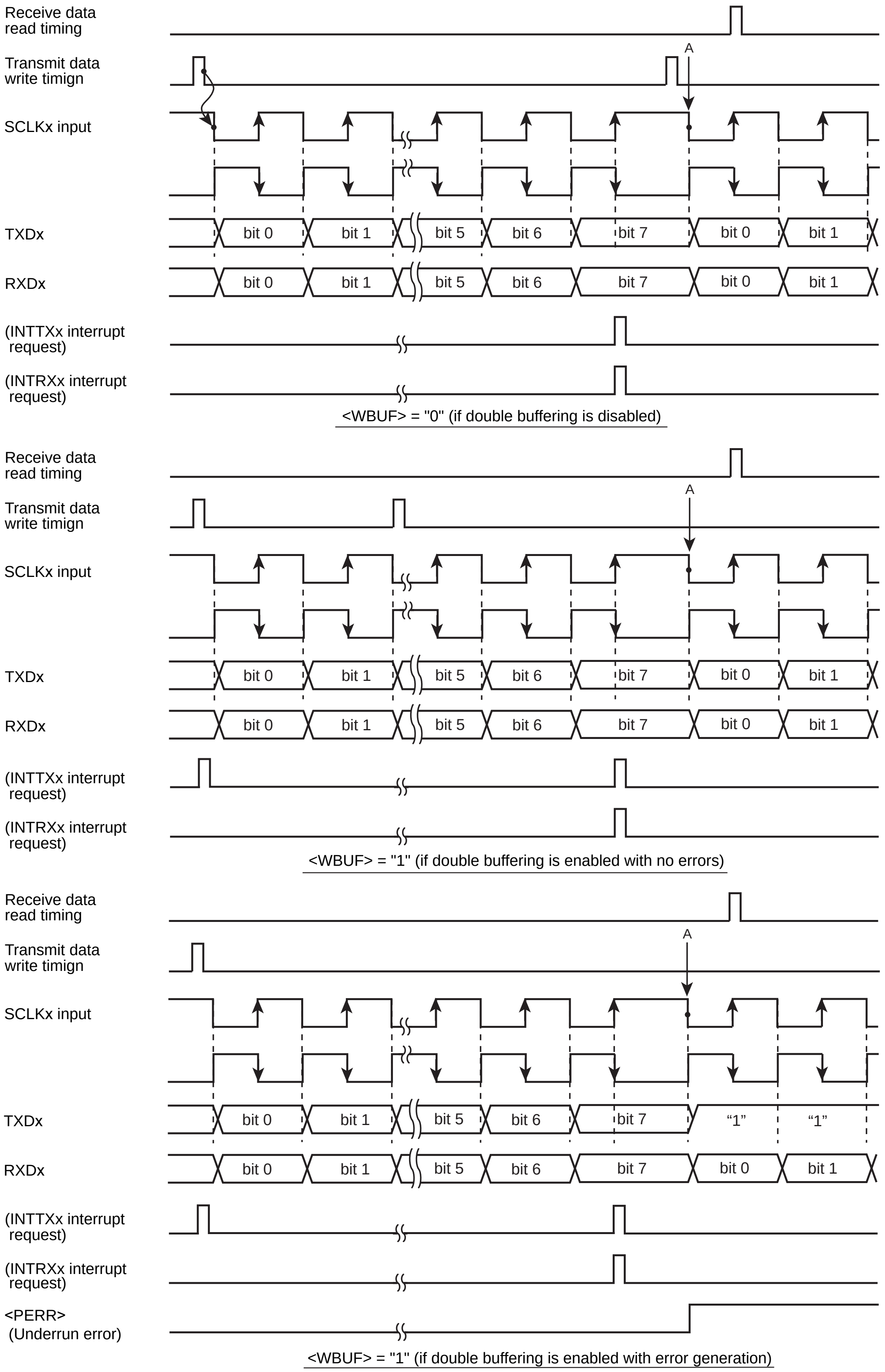
  Note that transmit data must be written into the transmit buffer before the SCLKx input for the next frame (data must be written before the point A in Figure 10-17). Data must be read before completing reception of the next frame data.
- If SCxMOD2<WBUF> is set to "1" and the double buffer is enabled.

  The interrupt INTTXx is generated at the timing when the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx interrupt is generated.

  Note that transmit data must be written into the transmit buffer before the SCLKx input for the next frame (data must be written before the point A in Figure 11-17). Data must be read before completing reception of the next frame data.

  Upon the SCLKx input for the next frame, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

  If data in receive buffer has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to transmit buffer when SCLKx for the next frame is input, an under-run error occurs and the dummy data (0xFF) is output.



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

<WBUF> = "1" (if double buffering is enabled with no errors)

<WBUF> = "1" (if double buffering is enabled with error generation)

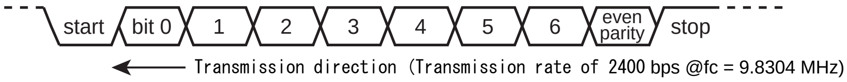
Figure 11-17 Transmit / Receive Operation in the I/O Interface Mode (SCLK Input Mode)

11.17.2 7-bit UART Mode

The 7-bit UART mode can be selected by setting the 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.


start
bit 0
1
2
3
4
5
6
even parity
stop
Transmission direction (Transmission rate of 2400 bps @fc = 9.8304 MHz)

Clocking condition	System clock :	High-speed (fc)
	High-speed clock gear :	x1 (fc)
	Prescaler clock :	fperiph/2 (fperiph = fsys)

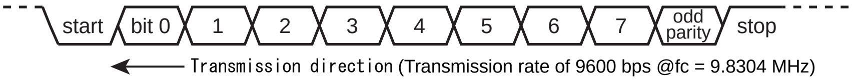
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x : don’t care - : no change

11.17.3 8-bit UART Mode

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10". In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows :


start
bit 0
1
2
3
4
5
6
7
odd parity
stop
Transmission direction (Transmission rate of 9600 bps @fc = 9.8304 MHz)

Clocking condition	System clock :	High-speed (fc)
	High-speed clock gear :	x1 (fc)
	Prescaler clock :	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	Set 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x : don't care - : no change

11.17.4 9-bit UART Mode

The 9-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "11". In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to SCxMOD0<TB8> for transmitting data. The received data is stored in SCxCR<RB8>.

When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SCxBUF. The stop bit length can be specified using SCxMOD2<SBLEN>.

11.17.4.1 Wake-up Function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SCxMOD0<WU> to "1".

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: **The TXDx pin of the slave controller must be set to the open drain output mode using the PxOD register.**

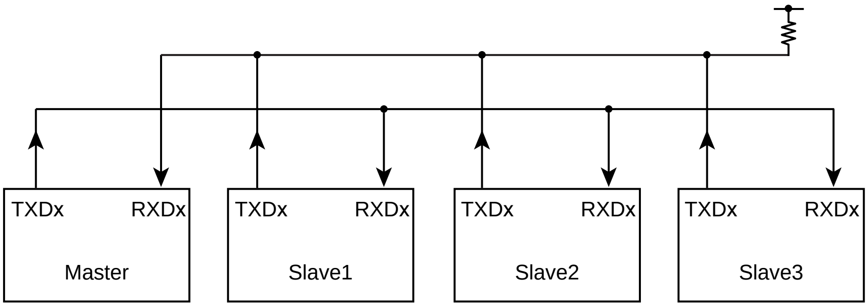


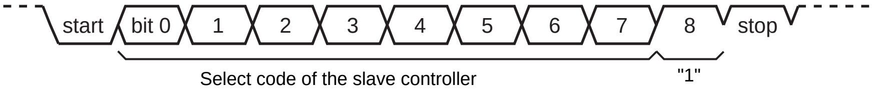
Figure 11-18 Serial Links to Use Wake-up Function

11.17.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

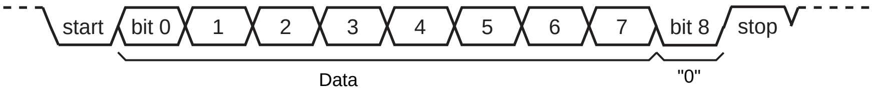
2. Set SCxMOD0<WU> to "1" for the slave controllers to make them ready to receive data.

3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller’s own select code, it clears the <WU> bit to "0".

5. The master controller transmits data to the designated slave controller (the controller of which <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 TMPM3U6FY/FW contains 2 Serial Bus Interface (I2C/SIO) channels, in which the following two oper-ating 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 Reg-ister	Port Output Control Register	Port Input Control Register	Port Open Drain Output Control Register
SBI0	I2C bus mode	SCL0 :PC1 SDA0 :PC0	PCFR3[1:0] = 11	PCCR[1:0] = 11	PCIE[1:0] = 11	PCOD[1:0] = 11
	SIO mode	SCK0 :PC2 SI0 :PC1 SO0 :PC0	PCFR3[2:0] = 111	PCCR[2:0] = 101(SCK0 output) PCCR[2:0] = 001(SCK0 input)	PCIE[2:0] = 010(SCK0 output) PCIE[2:0] = 110(SCK0 input)	PCOD[2:0] = xxx
SBI1	I2C bus mode	SCL1 :PG1 SDA1 :PG0	PGFR3[1:0] = 11	PGCR[1:0] = 11	PGIE[1:0] = 11	PGOD[1:0] = 11
	SIO mode	SCK1 :PG2 SI1 :PG1 SO1 :PG0	PGFR3[2:0] = 111	PGCR[2:0] = 101(SCK1 output) PGCR[2:0] = 001(SCK1 input)	PGIE[2:0] = 010(SCK1 output) PGIE[2:0] = 110(SCK1 input)	PGOD[2:0] = xxx

Note:x: Don't care

12.1 Configuration

The configuration is shown in Figure 12-1.

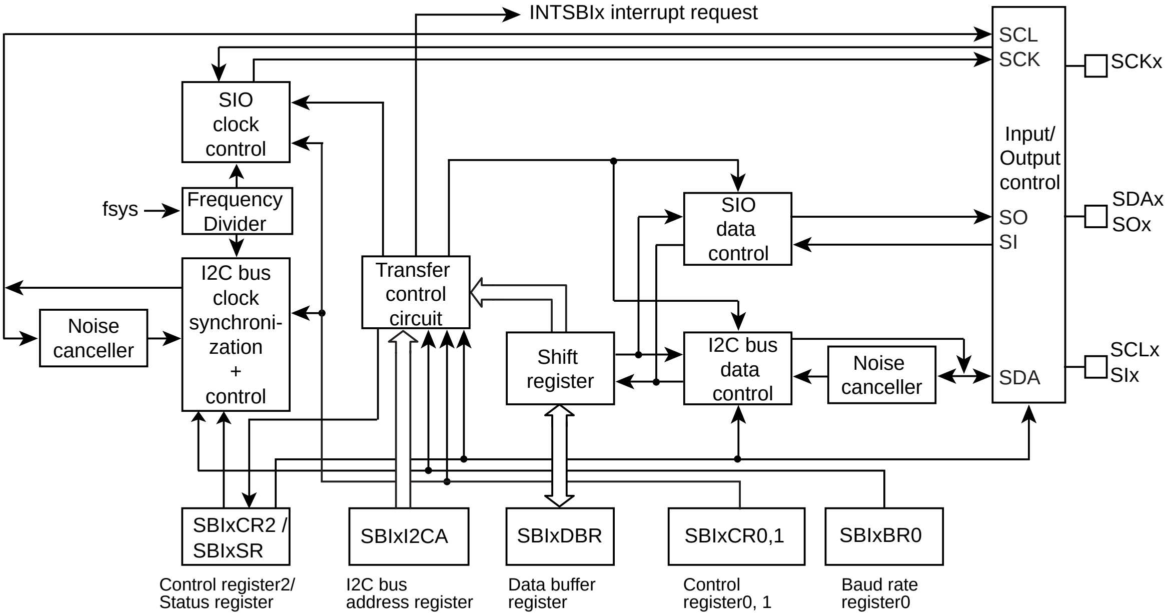


Figure 12-1 (I2C/SIO) Block Interface

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

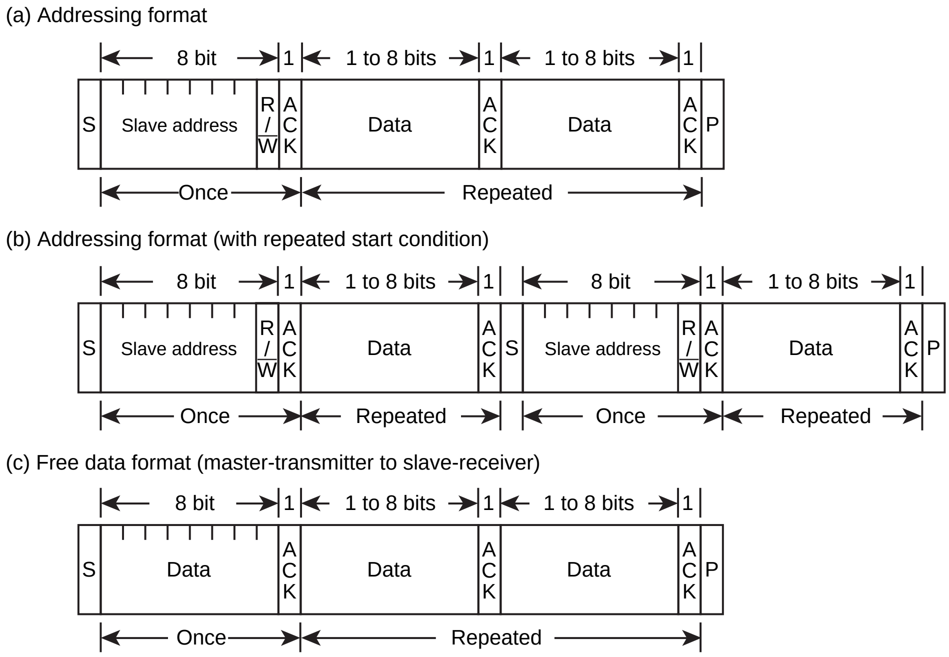
The tables below show the registers and register addresses for each channel.

Channel x	Base Address
Channel0	0x4002_0000
Channel1	0x4002_0020

Register name(x=0,1,)		Address(Base+)
Control register 0	SBIxCR0	0x0000
Control register 1	SBIxCR1	0x0004
Data buffer register	SBIxDBR	0x0008
I2C bus address register	SBIxI2CAR	0x000C
Control register 2	SBIxCR2 (writing)	0x0010
Status register	SBIxSR (reading)	
Baud rate register 0	SBIxBR0	0x0014

12.3 I2C Bus Mode Data Format

Figure 12-2 shows the data formats used in the I2C bus mode.



(a) Addressing format

(b) Addressing format (with repeated start condition)

(c) Free data format (master-transmitter to slave-receiver)

Note) S : Start condition
R/W : Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 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 SBIxCR0(Control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBIxCR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it’s been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

Note: To use the serial bus interface, enable this bit first.

12.4.2 SBIxCR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
After reset	0	0	0	0	1	0	0	1(Note3)

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>-</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7-5</td><td>BC[2:0]</td><td>R/W</td><td>Select the number of bits per transfer (Note 1)(Note 6)
<table>
<tr><th rowspan="2">&lt;BC&gt;</th><th colspan="2">When &lt;ACK&gt; = 0</th><th colspan="2">When &lt;ACK&gt; = 1</th></tr>
<tr><th>Number of clock cycles</th><th>Data length</th><th>Number of clock cycles</th><th>Data length</th></tr>
<tr><td>000</td><td>8</td><td>8</td><td>9</td><td>8</td></tr>
<tr><td>001</td><td>1</td><td>1</td><td>2</td><td>1</td></tr>
<tr><td>010</td><td>2</td><td>2</td><td>3</td><td>2</td></tr>
<tr><td>011</td><td>3</td><td>3</td><td>4</td><td>3</td></tr>
<tr><td>100</td><td>4</td><td>4</td><td>5</td><td>4</td></tr>
<tr><td>101</td><td>5</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>110</td><td>6</td><td>6</td><td>7</td><td>6</td></tr>
<tr><td>111</td><td>7</td><td>7</td><td>8</td><td>7</td></tr>
</table></td></tr>
<tr><td>4</td><td>ACK</td><td>R/W</td><td>Master mode
0: Acknowledgement clock pulse is not generated.
1: Acknowledgement clock pulse is generated.
Slave mode
0: Acknowledgement clock pulse is not counted.
1: Acknowledgement clock pulse is counted.</td></tr>
<tr><td>3</td><td>-</td><td>R</td><td>Read as 1.</td></tr>
<tr><td>2-1</td><td>SCK[2:1]</td><td>R/W</td><td rowspan="2">Select internal SCL output clock frequency (Note 2)(Note 7).
<table>
<tr><td>000</td><td>n = 5</td><td>385 kHz</td><td rowspan="8">System Clock: fsys
(=40MHz)
Clock gear : fc/1
Frequency = $\frac{fsys}{2^n + 72}$ [Hz]</td></tr>
<tr><td>001</td><td>n = 6</td><td>294 kHz</td></tr>
<tr><td>010</td><td>n = 7</td><td>200 kHz</td></tr>
<tr><td>011</td><td>n = 8</td><td>122 kHz</td></tr>
<tr><td>100</td><td>n = 9</td><td>68 kHz</td></tr>
<tr><td>101</td><td>n = 10</td><td>36 kHz</td></tr>
<tr><td>110</td><td>n = 11</td><td>19 kHz</td></tr>
<tr><td>111</td><td></td><td>reserved</td></tr>
</table></td></tr>
<tr><td rowspan="2">0</td><td>SCK[0]</td><td>W</td></tr>
<tr><td>SWRMON</td><td>R</td><td>On reading <SWRMON>: Software reset status monitor
0:Software reset operation is in progress.
1:Software reset operation is not in progress.</td></tr>
</table>

Note 1: **Clear <BC[2:0]> to "000" before switching the operation mode to the SIO mode.**

Note 2: **For details on the SCL line clock frequency, refer to "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 SBIxCR2 register, the initial value of the <SCK[0]> bit is "0".**

Note 4: **The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.**

Note 5: **When <BC[2:0]>="001" and <ACK>="0" in master mode, SCL line may be fixed to "L" by falling edge of SCL line after generation of STOP condition and the other master devices can not use the bus. In the case of bus which is connected with several master devices, the number of bits per transfer should be set equal or more than 2 before generation of STOP condition.**

Note 6: When rewriting under communication, please perform it in the period from interrupt generating after an address or data transfer to the return from interrupt.

Note 7: Do not rewrite this during communication.

12.4.3 SBIxCR2(Control register 2)

This register serves as SBIxSR register by reading it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. 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 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBIx interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: − 1:Detected
2	AAS	R	Slave address match detection 0: − 1: Detected (This bit is set when the general-call address is detected as well.)
1	AD0	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

12.4.5 SBIxBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

12.4.6 SBIxDBR (Serial bus interface data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R (Receive)/ W (Transmit)	Receive data / Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

12.4.7 SBIxI2CAR (I2Cbus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: **Please set the bit 0 <ALS> of I2C bus address register SBIxI2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.**

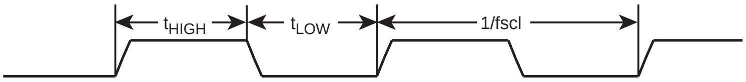
Note 2: **Do not set SBIxI2CAR to "0x00" in slave mode. (If SBIxI2CAR is set to "0x00", it's recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

12.5 Control in the I2C Bus Mode

12.5.1 Serial Clock

12.5.1.1 Clock source

SBIxCR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$$t_{LOW} = 2^{n-1}/fsys + 58/fsys$$
$$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$$
$$fscl = 1/(t_{LOW} + t_{HIGH})$$
$$= \frac{fsys}{2^n + 72}$$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 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.

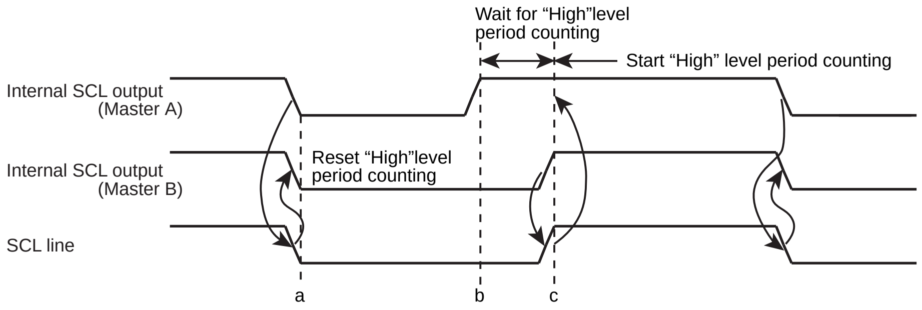

Wait for "High"level period counting
Start "High" level period counting
Internal SCL output (Master A)
Reset "High"level period counting
Internal SCL output (Master B)
SCL line
a
b
c

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 SBIxCR1<ACK> to "1" selects the acknowledge mode. When operating as a master, the SBI adds one clock for acknowledgment signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the SBI releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. 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

SBIxCR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

12.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBIxI2CAR<ALS> and a slave address in SBIxI2CAR<SA[6:0]> sets addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

12.5.5 Operating mode

The setting of SBIxCR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, ensure that the serial bus interface pins are at "High" level before setting <SBIM[1:0]> to "10". Also, ensure that the bus is free before switching the operating mode to the port mode.

12.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2<TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode:

- When data is transmitted in the addressing format.
- When the received slave address matches the value specified at SBIxI2CAR.
- When a general-call address is received; i.e., the eight bits following the start condition are all zeros.

If the value of the direction bit (R/$\overline{\text{W}}$) is "1", <TRX> is set to "1" by the hardware. If the bit is "0", <TRX> is set to "0".

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0", <TRX> changes to "1". If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

If SBI is used in free data format, <TRX> is not changed by the hardware.

12.5.7 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

12.5.8 Generating Start and Stop Conditions

When SBIxSR<BB> is "0", writing "1" to SBIxCR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

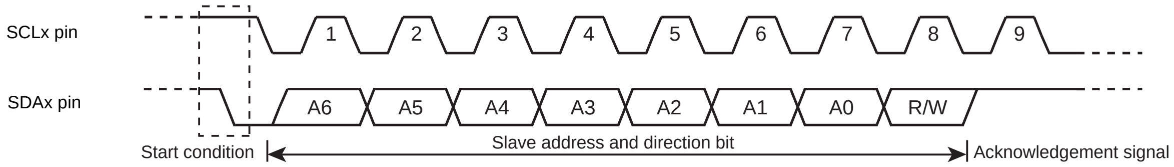


Figure 12-5 Generating the Start Condition and a Slave Address

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.

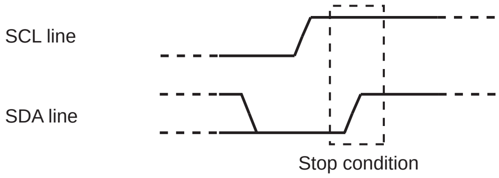


Figure 12-6 Generating the Stop Condition

SBIxSR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and cleared to "0" when the stop condition is detected (the bus is free).

12.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBIx) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBIx is generated under the following conditions.

- After output of the acknowledge signal which is generated when the received slave address matches the slave address set to SBIxI2CAR<SA[6:0]>.
- After the acknowledge signal is generated when a general-call address is received.
- When the slave address matches or a data transfer is completed after receiving a general-call address.

In the address recognition mode (<ALS> = "0"), INTSBIx is generated when the received slave address matches the values specified at SBIxI2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBIx) is generated, SBIxCR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the SBI pulls the SCL line to the "Low" level.

<PIN> is set to "1" when data is written to or read from SBIxDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1". When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note: When arbitration occurs while a slave address and direction bit are transferred in the master mode, <PIN> is cleared to "0" and INTSBIx occurs. This does not relate to whether a slave address matches <SA>.

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.

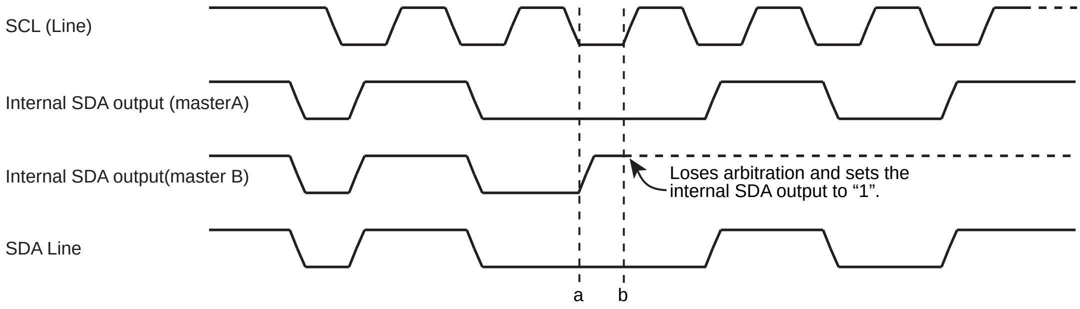

SCL (Line)
Internal SDA output (masterA)
Internal SDA output(master B)
SDA Line
Loses arbitration and sets the internal SDA output to “1”.
a
b

Figure 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 SBIxSR<AL> is set to "1".

When an arbitration lost occurs, SBIxSR<MST> and <TRX> are cleared to "0", causing the SBI to operate as a slave receiver and it stops the clock output during data transfer. If the master device which sends a slave address and direction bit generates Arbitration lost, it receives a slave address and direction bit which are sent by other master devices as slave device. Regardless of whether a received slave address matches <SA>, <PIN> is cleared to "0" and INTSBIx occurs.

<AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.

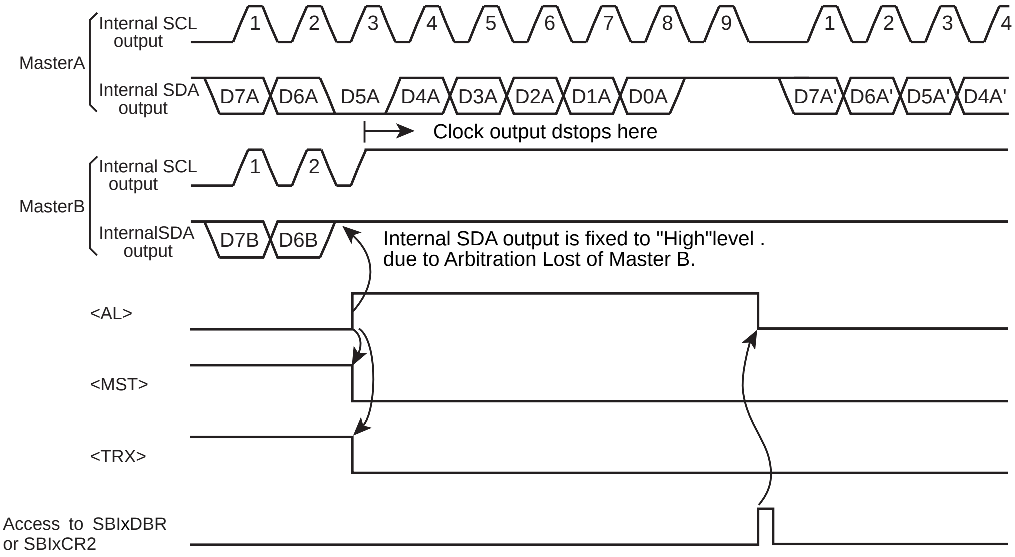


Figure 12-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

12.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBIxI2CAR<ALS>="0"), SBIxSR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBIxI2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIxDBR.

12.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR<AD0> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<AD0> is cleared to "0" when the start or stop condition is detected on the bus.

12.5.13 Last Received Bit Monitor

SBIxSR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBIxSR<LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

12.5.14 Data Buffer Register (SBIxDBR)

Reading or writing SBIxDBR initiates reading received data or writing transmitted data.

When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

12.5.15 Baud Rate Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

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 SBIxCR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. When writing SBIxCR2<SWRST[1:0]>, set SBIxCR2<MST><TRX><BB><PIN> to "0000" and SBIxCR2<SBIM[1:0]> to "10" for I2C bus mode. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to "0".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

12.6 Data Transfer Procedure in the I2C Bus Mode

12.6.1 Device Initialization

Firstly, set SBIxCR1<ACK><SCK[2:0]>. Set "1" to <ACK> to specify the acknowledgement mode. Set "000" to SBIxCR1<BC[2:0]>.

Secondly, set <SA[6:0]> (a slave address) and <ALS> to SBIxI2CAR. (In the addressing format mode, set <ALS>="0").

Finally, to configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "000" to SBIxCR2<MST><TRX><BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "00" to <SWRST[1:0]>.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	0	1	0	X	X	X	Specifies ACK and SCL clock.
SBIxI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBIxCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don’t care

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 SBIxCR1<ACK> to select the acknowledgment mode. Write to SBIxDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBIxCR2<MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIxDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the master mode, the SBI holds the SCL line at the "Low" level while <PIN> is = "0".<TRX> changes its value according to the transmitted direction bit at generation of the INTSBIx interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIxDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

```
                        7   6   5   4   3   2   1   0
        Reg.      ←    SBIxSR
        Reg.      ←    Reg. e 0x20
        if Reg.   ≠    0x00                                  Ensures that the bus is free.

        Then

        SBIxCR1   ←    X   X   X   1   0   X   X   X         Selects the acknowledgement mode.
        SBIxDBR   ←    X   X   X   X   X   X   X   X         Specifies the desired slave address and direction.
        SBIxCR2   ←    1   1   1   1   1   0   0   0         Generates the start condition.
                  Example of INTSBI0 interrupt routine
                       Clears the interrupt request.
                       Processing
                       End of interrupt
```

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 SBIxI2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "Low" level during the ninth clock and outputs an acknowledgment signal.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the slave mode, the SBI holds the SCL line at the "Low" level while <PIN> is "0".

Note: **The user can only use a DMA transfer:**

- **When there is only one master and only one slave and**
- **Continuous transmission or reception is possible.**

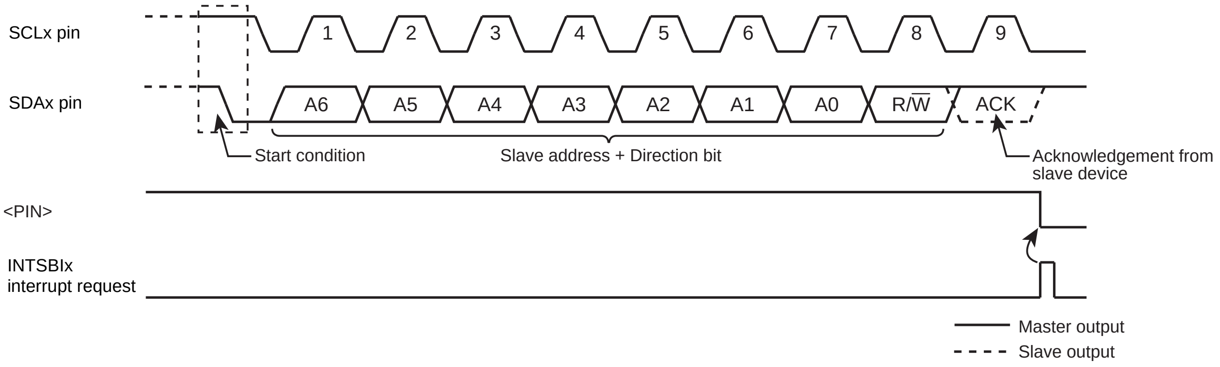


Figure 12-9 Generation of the Start Condition and a Slave Address

12.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBIx 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 SBIxDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIxDBR. Writing the data makes <PIN> to "1", causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word.

After the transfer is completed, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIx interrupt

```
if MST = 0
Then go to the slave-mode processing.
if TRX = 0
Then go to the receiver-mode processing.
if LRB = 0
Then go to processing for generating the stop condition.
SBIxCR1    ←  X  X  X  X  0  X  X  X     Specifies the number of bits to be transmitted and
                                         specify whether ACK is required.
SBIxDBR    ←  X  X  X  X  X  X  X  X     Writes the transmit data.
End of interrupt processing.
```

Note: **X; Don’t care**

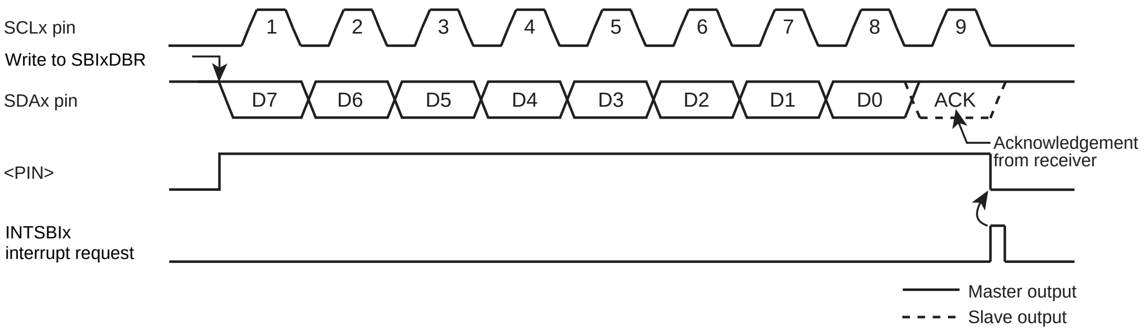


Figure 12-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIxDBR.

If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIxDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)On reading the data, <PIN> is set to "1", and the serial clock is output to the SCL pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SDA pin.

After that, the INTSBIx interrupt request is generated, and <PIN> is cleared to "0", pulling the SCL pin to the "Low" level. Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.

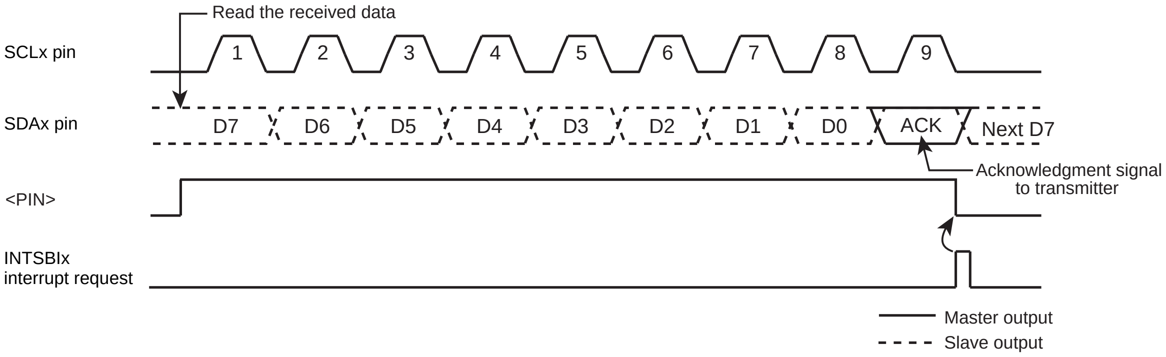


Figure 12-11 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

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.

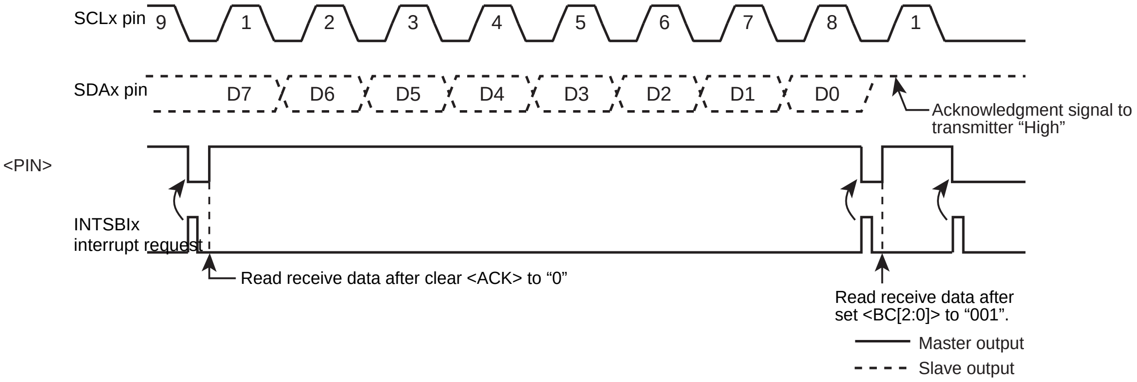

SCLx pin
9
1
2
3
4
5
6
7
8
1
SDAx pin
D7
D6
D5
D4
D3
D2
D1
D0
Acknowledgment signal to transmitter "High"
<PIN>
INTSBIx interrupt request
Read receive data after clear <ACK> to "0"
Read receive data after set <BC[2:0]> to "001".
Master output
Slave output

Figure 12-12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIx interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIxDBR								Reads dummy data.

End of interrupt

INTSBIx interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxDBR								Reads the first to (N-2)th data words.

End of interrupt

INTSBIx interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the (N-1)th data word.

End of interrupt

INTSBIx interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the Nth data word.

End of interrupt

INTSBIx interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.

End of interrupt

Note: X; Don't care

12.6.3.2 Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBIx interrupt request on four occasions:

1) When the SBI has received any slave address from the master.

2) When the SBI has received a general-call address.

3) When the received slave address matches its address.

4) When a data transfer has been completed in response to a general-call.

Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode.

Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

When data is written to or read from SBIxDBR or when <PIN> is set to "1", the SCLx pin is released after a period of t_{LOW}.

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.

SBIxSR<AL>, <TRX>, <AAS> and <AD0> 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.

INTSBIx interrupt

```
if TRX = 0
Then go to other processing.
if AL = 0
Then go to other processing.
if AAS = 0
Then go to other processing.
SBIxCR1    ←  X  X  X  1  0  X  X  X      Sets the number of bits to be transmitted.
SBIxDBR    ←  X  X  X  X  X  X  X  X      Sets the transmit data.
```

Note: X; Don’t care

Table 12-2 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC[2:0]> and write the transmit data into SBIxDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIxDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC[2:0].> and read the received data from SBIxDBR.

12.6.4 Generating the Stop Condition

When SBIxSR<BB> is "1", writing "1" to SBIxCR2<MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus.

Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released.

After that, the SDA pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

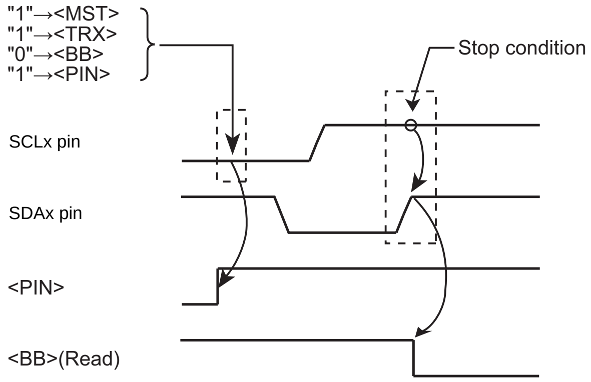


Figure 12-13 Generating the Stop Condition

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 SBIxCR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy.

Then, test SBIxSR<BB> and wait until it becomes "0" to ensure that the SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCLx bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "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	
SBIxCR2	←	0	0	0	1	1	0	0	0	Releases the bus.
if SBIxSR<BB> ≠ 0										Checks that the SCL pin is released.
Then										
if SBIxSR<LRB> ≠ 1										Checks that no other device is pulling the SCL pin to the "Low".
Then										
4.7 μs Wait										
SBIxCR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgment mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Sets the desired slave address and direction.
SBIxCR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Note: X; Don’t care

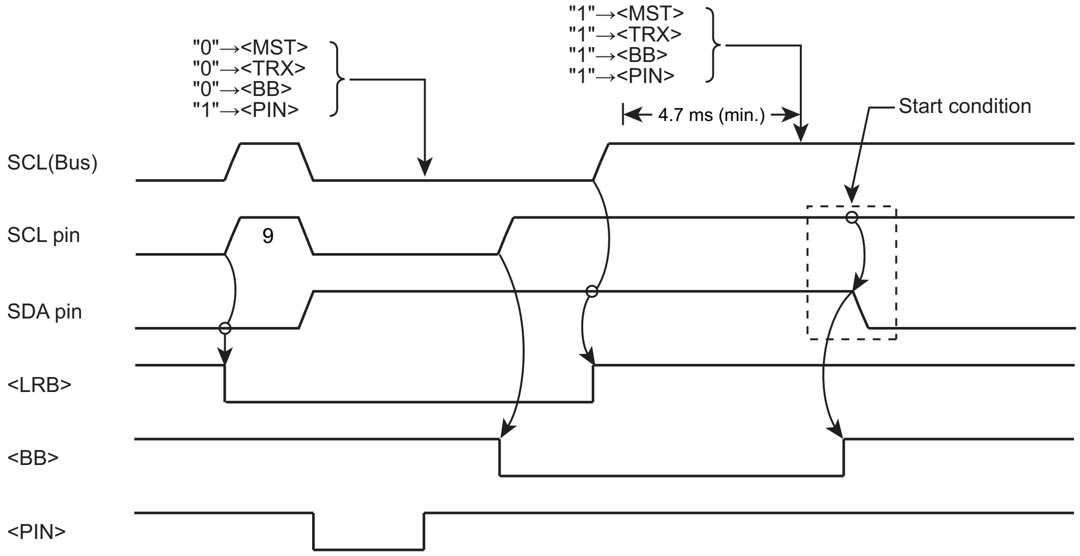


Figure 12-14 Timing Chart of Generating a Restart

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 (INTSBIx) 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 SBIxCR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBIxCR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

12.8.2 SBIxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1) 000 n = 3 2.5 MHz 001 n = 4 1.25 MHz 010 n = 5 625 kHz 011 n = 6 313 kHz 100 n = 7 156 kHz 101 n = 8 78 kHz 110 n = 9 39 kHz 111 − External clock System clock: fsys (=40MHz) Clock gear: fc/1 Frequency = $\frac{fsys/2}{2^n}$ [Hz]

Note 1: **After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBIxCR2 register and the SBIxSR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

12.8.3 SBIxDBR (Data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIxDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

12.8.4 SBIxCR2(Control register 2)

This register serves as SBIxSR register by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

Note 2: **Make sure that modes are not changed during a communication session.**

12.8.5 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	-	R	Read as 1.(Note 1)
3	SIOF	R	Serial transfer status monitor. 0: Completed 1: In progress
2	SEF	R	Shift operation status monitor 0: Completed. 1: In progress
1-0	-	R	Read as 1. (Note 1)

Note 1: **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 SBIxBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

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 SBIxCR1<SCK[2:0]>.

(1) Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

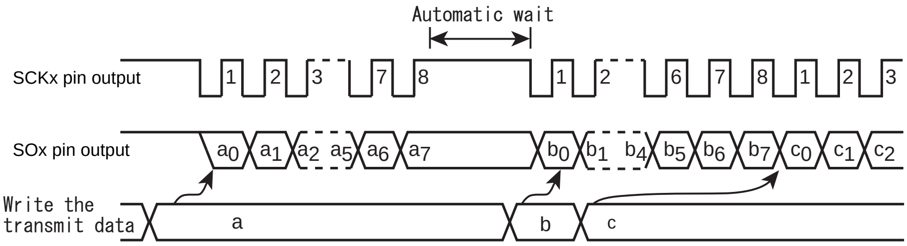


Figure 12-15 Automatic Wait

(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.

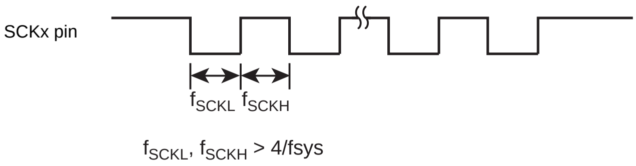


Figure 12-16 Maximum Transfer Frequency of External Clock Input

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).

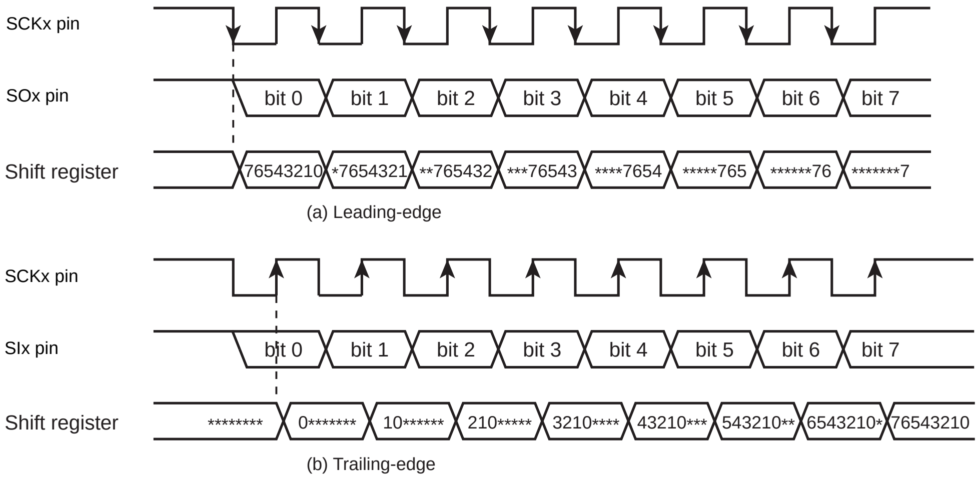


Figure 12-17 Shift Edge

12.9.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1<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 SBIxDBR.

After writing the transmit data, writing "1" to SBIxCR1<SIOS> starts the transmission. The transmit data is moved from SBIxDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIxDBR becomes empty, and the INTSBIx (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIxDBR is loaded with the next transmit data.

In the external clock mode, SBIxDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIxDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBIxSR<SIOF> to "1" to the falling edge of SCK.

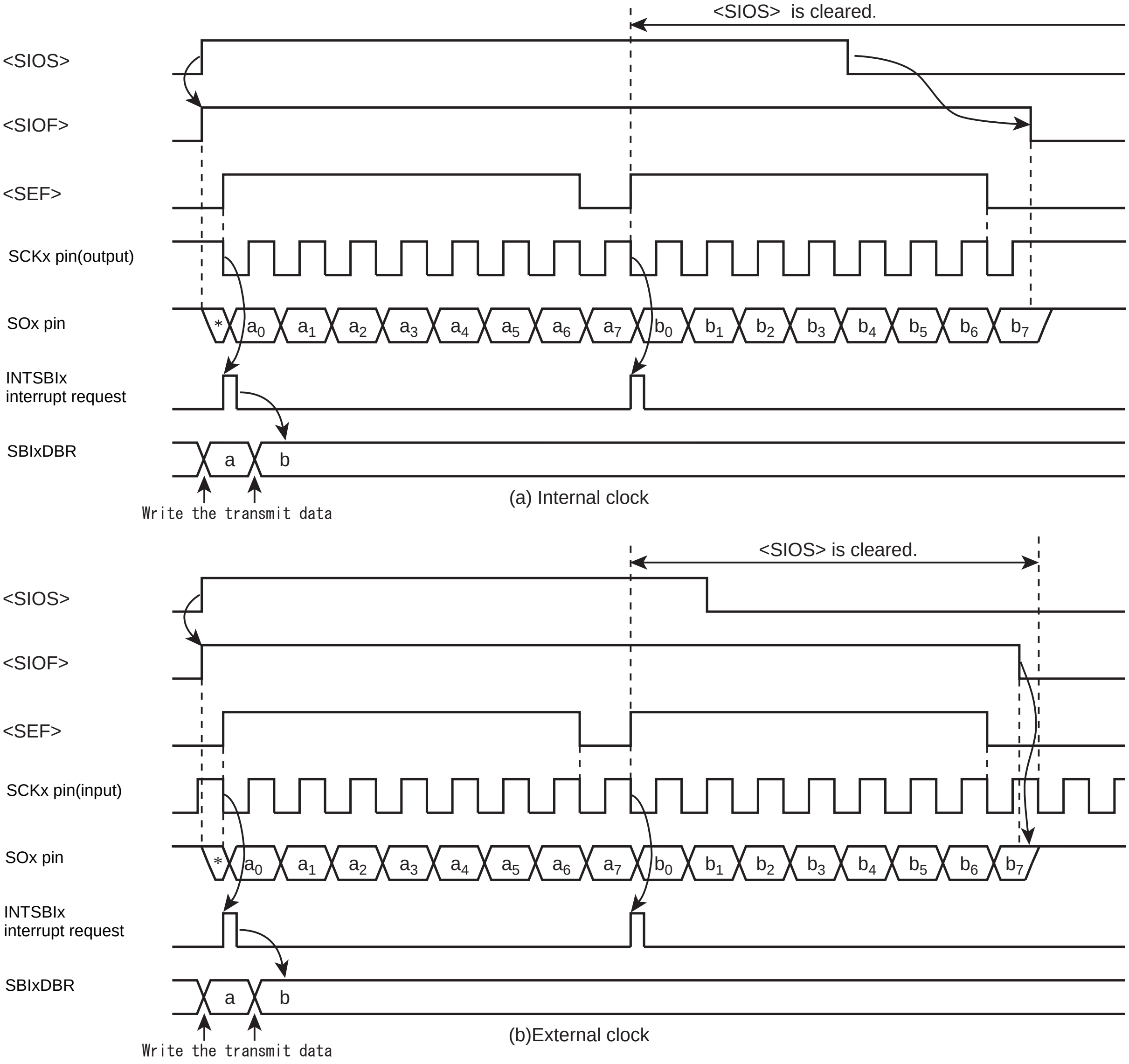
Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBIxSR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBIx interrupt

SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.



(a) Internal clock

(b)External clock

Figure 12-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

		7	6	5	4	3	2	1	0	
if SBIxSR<SIOF> ≠ 0										Recognizes the completion of the transmission.
Then										
if SCK ≠ 1										Recognizes "1" is set to the SCK pin by monitoring the port.
Then										
SBIxCR1	←	0	0	0	0	0	1	1	1	Completes the transmission by setting <SIOS> = 0.

12.9.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBIxCR1<SIOS> enables reception. Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIxDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIxDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBIxCR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBIx interrupt

Reg.	←	SBIxDBR	Reads the received data.

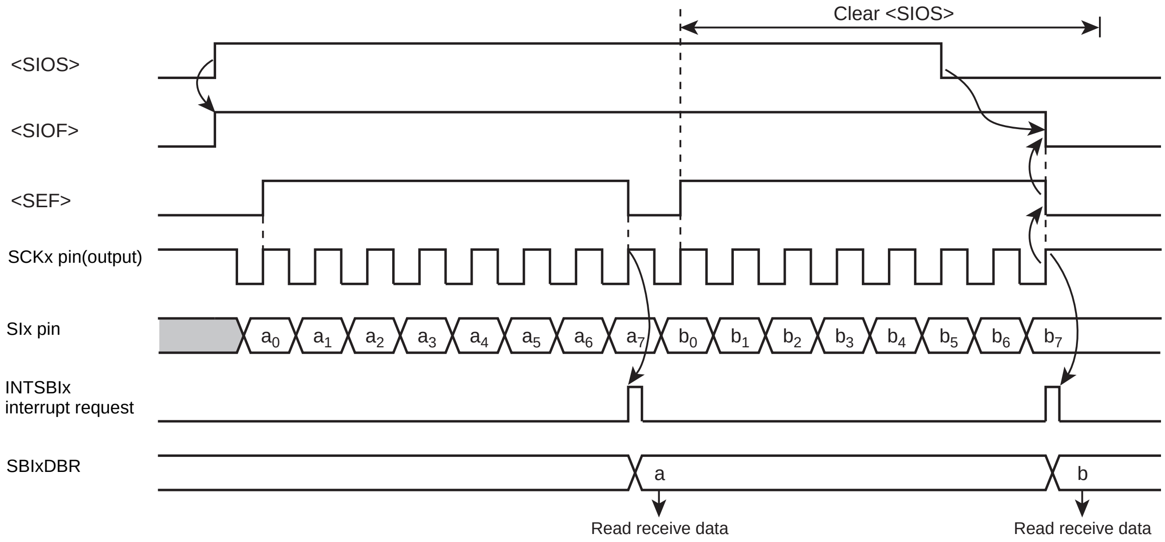


Figure 12-19 Receive Mode (Example: Internal Clock)

12.9.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIxDBR and setting SBIxCR1<SIOS> to "1" enables transmission and reception.The transmit data is output through the SOx pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIxDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started.The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBIxCR1<SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception. If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: **The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

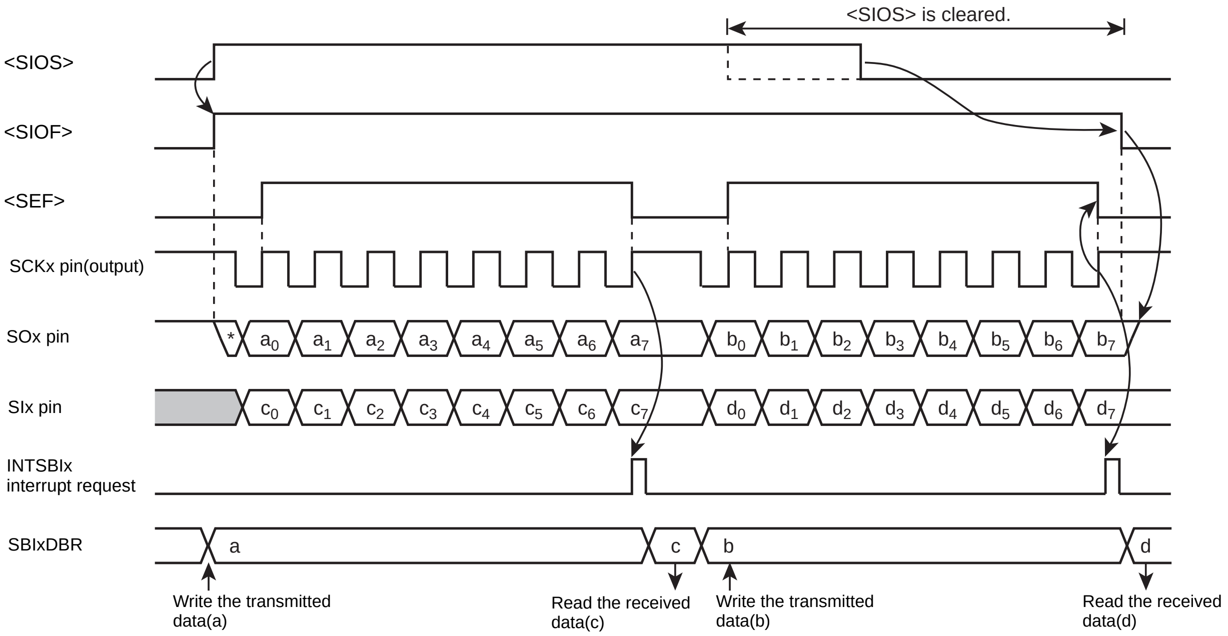

<SIOS> is cleared.
<SIOS>
<SIOF>
<SEF>
SCKx pin(output)
SOx pin
* a0 a1 a2 a3 a4 a5 a6 a7 b0 b1 b2 b3 b4 b5 b6 b7
SIx pin
c0 c1 c2 c3 c4 c5 c6 c7 d0 d1 d2 d3 d4 d5 d6 d7
INTSBIx
interrupt request
SBIxDBR
a
c
b
d
Write the transmitted data(a)
Read the received data(c)
Write the transmitted data(b)
Read the received data(d)

Figure 12-20 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBIx interrupt

Reg.	←	SBIxDBR								Reads the received data.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

12.9.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBIxCR1<SIOS>= "0", the last bit of the transmitted data retains the data of SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

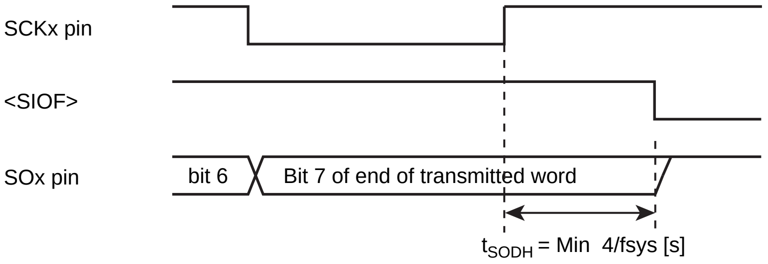


Figure 12-21 Data retention time of the last bit at the end of transmission

13. Synchronous Serial Port (SSP)

13.1 Overview

This LSI contains the SSP (Synchronous Serial Port) with 2 channels. These channels have the following features.

Communication protocol		Three types of synchronous serial ports including the SPI ・ Motorola SPI (SPI) frame format ・ TI synchronous (SSI) frame format ・ National Microwire (Microwire) frame format
Operation mode		Master/slave mode
Transmit FIFO		16bits wide / 8 tiers deep
Receive FIFO		16bits wide / 8 tiers deep
Transmitted/received data size		4 to 16 bits
Interrupt type		Transmit interrupt Receive interrupt Receive overrun interrupt Time-out interrupt
Communication speed	In master mode	φT0 / 2 (max. 10Mbps)
	In slave mode	φT0 (40MHz)/ 12 (max. 3.3Mbps)
DMA		Supported
Internal test function		The internal loopback test mode is available.
Control pin (x = 0 to 1)		SPxCLK,SPxFSS,SPxDO,SPxDI

13.2 Block Diagram

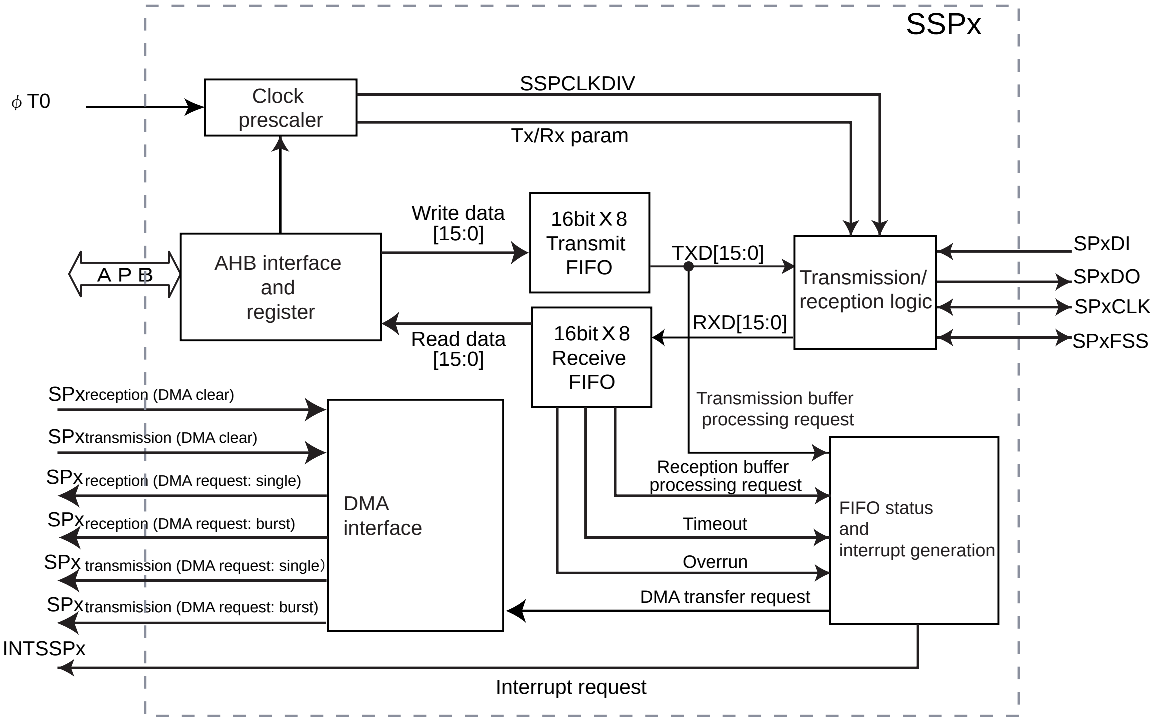

SSPx
ϕ T0
Clock prescaler
SSPCLKDIV
Tx/Rx param
Write data [15:0]
16bit X 8 Transmit FIFO
TXD[15:0]
APB
AHB interface and register
Transmission/ reception logic
SPxDI
SPxDO
SPxCLK
SPxFSS
Read data [15:0]
16bit X 8 Receive FIFO
RXD[15:0]
SPxreception (DMA clear)
SPxtransmission (DMA clear)
SPx reception (DMA request: single)
SPxreception (DMA request: burst)
SPx transmission (DMA request: single)
SPxtransmission (DMA request: burst)
DMA interface
Transmission buffer processing request
Reception buffer processing request
Timeout
Overrun
DMA transfer request
FIFO status and interrupt generation
INTSSPx
Interrupt request

Figure 13-1 SSP block diagram

13.3 Register

13.3.1 Register List

Channel x	Base Address
Channel0	0x400C_0000
Channel1	0x400C_1000

Register Name(x=0 to 1)		Address (Base+)
Control register 0	SSPxCR0	0x0000
Control register 1	SSPxCR1	0x0004
Receive FIFO (read) and transmit FIFO (write) data register	SSPxDR	0x0008
Status register	SSPxSR	0x000C
Clock prescale register	SSPxCPSR	0x0010
Interrupt enable/disable register	SSPxIMSC	0x0014
Pre-enable interrupt status register	SSPxRIS	0x0018
Post-enable interrupt status register	SSPxMIS	0x001C
Interrupt clear register	SSPxICR	0x0020
DMA control register	SSPxDMACR	0x0024
Reserved	-	0x0028 to 0x0FFC

Note 1: These registers in the above table allows to access only word (32 bits) basis.

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

13.3.2 SSPxCR0(Control register 0)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-16</td><td>−</td><td>W</td><td>Write as "0".</td></tr>
<tr><td>15-8</td><td>SCR[7:0]</td><td>R/W</td><td>For serial clock rate setting.
Parameter : 0x00 to 0xFF.

Bits to generate the SSP transmit bit rate and receive bit rate.
This bit rate can be obtained by the following equation.
Bit rate = φT0/ (<CPSDVSR> × (1+ <SCR>))
<CPSDVSR> is an even number between 2 to 254, which is programmed by the SSPxCPSR register, and <SCR> takes a value between 0 to 255.</td></tr>
<tr><td>7</td><td>SPH</td><td>R/W</td><td>SPxCLK phase:
0 : Captures data at the 1st clock edge.
1 : Captures data at the 2nd clock edge.
This is applicable to Motorola SPI frame format only. Refer to Section "Motorola SPI frame format"</td></tr>
<tr><td>6</td><td>SPO</td><td>R/W</td><td>SPxCLK polarity:
0:SPxCLK is in Low state.
1:SPxCLK is in High state.
This is applicable to Motorola SPI frame format only. Refer to Section "Motorola SPI frame format"</td></tr>
<tr><td>5-4</td><td>FRF[1:0]</td><td>R/W</td><td>Frame format:
00: SPI frame format
01: SSI serial frame format
10: Microwire frame format
11: Reserved, undefined operation</td></tr>
<tr><td>3-0</td><td>DSS[3:0]</td><td>R/W</td><td>Data size select:
<table>
<tr><td>0000:</td><td>Reserved, undefined operation</td><td>1000:</td><td>9 bits data</td></tr>
<tr><td>0001:</td><td>Reserved, undefined operation</td><td>1001:</td><td>10 bits data</td></tr>
<tr><td>0010:</td><td>Reserved, undefined operation</td><td>1010:</td><td>11 bits data</td></tr>
<tr><td>0011:</td><td>4 bits data</td><td>1011:</td><td>12 bits data</td></tr>
<tr><td>0100:</td><td>5 bits data</td><td>1100:</td><td>13 bits data</td></tr>
<tr><td>0101:</td><td>6 bits data</td><td>1101:</td><td>14 bits data</td></tr>
<tr><td>0110:</td><td>7 bits data</td><td>1110:</td><td>15 bits data</td></tr>
<tr><td>0111:</td><td>8 bits data</td><td>1111:</td><td>16 bits data</td></tr>
</table></td></tr>
</table>

Note: Set a clock prescaler to SSPxCR0<SCR[7:0]> = 0x00 , SSPxCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

13.3.3 SSPxCR1(Control register1)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	SOD	R/W	Slave mode SPxDO output control: 0: Enable 1: Disable Slave mode output disable. This bit is relevant only in the slave mode (<MS>="1").
2	MS	R/W	Master/slave mode select: (Note) 0: Device configured as a master. 1: Device configured as a slave.
1	SSE	R/W	SSP enable/disable 0: Disable 1: Enable
0	LBM	R/W	Loop back mode 0: Normal serial port operation enabled. 1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

Note: This bit is for switching between master and slave. Be sure to configure in the following steps in slave mode and in transmission.

1)	Set to slave mode	:<MS>=1
2)	Set transmit data in FIFO	:<DATA>=0x****
3)	Set SSP to Enable.	:<SSE>=1

13.3.4 SSPxDR(Data register)

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

Bit	Bit Symbol	Type	Function
31-16	−	W	Write as "0".
15-0	DATA[15:0]	R/W	Transmit/receive FIFO data: 0x0000 to 0xFFFF Read: Receive FIFO Write: Transmit FIFO If the data size used in the program is less than 16bits, write the data to fit LSB.The transmit control circuit ignores unused bits of MSB side. The receive control circuit receives the data to fit LSB automatically.

13.3.5 SSPxSR(Status register)

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

Bit	Bit Symbol	Type	Function
31-5	−	W	Write as "0".
4	BSY	R	Busy flag: 0: Idle 1: Busy <BSY>="1" indicates that the SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.
3	RFF	R	Receive FIFO full flag: 0: Receive FIFO is not full. 1: Receive FIFO is full.
2	RNE	R	Receive FIFO empty flag: 0: Receive FIFO is empty. 1: Receive FIFO is not empty.
1	TNF	R	Transmit FIFO full flag: 0: Transmit FIFO is full. 1: Transmit FIFO is not full.
0	TFE	R	Transmit FIFO empty flag: 0: Transmit FIFO is not empty. 1: Transmit FIFO is empty.

13.3.6 SSPxCPSR (Clock prescale register)

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

Bit	Bit Symbol	Type	Function
31-8	−	W	Write as "0".
7-0	CPSDVSR[7:0]	R/W	Clock prescale divisor: Set an even number from 2 to 254. Clock prescale divisor: Must be an even number from 2 to 254, depending on the frequency of φT0. The least significant bit always returns zero when read.

Note: Set a clock prescaler to SSPxCR0<SCR[7:0]> = 0x00 , SSPxCPSR<CPSDVSR[7:0]> = 0x02, when slave mode is selected.

13.3.7 SSPxIMSC (Interrupt enable/disable register)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	TXIM	R/W	Transmit FIFO interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to occur if the transmit FIFO is half empty or less.
2	RXIM	R/W	Receive FIFO interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to occur if the receive FIFO is half full or less.
1	RTIM	R/W	Receive time-out interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to indicate that data exists in the receive FIFO to the time-out period and data is not read.
0	RORIM	R/W	Receive overrun interrupt enable: 0: Disable 1: Enable Enable or disable a conditional interrupt to indicate that data was written when the receive FIFO was in the full condition.

13.3.8 SSPxRIS (Pre-enable interrupt status register)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	TXRIS	R	Pre-enable transmit interrupt flag: 0: Interrupt not present 1: Interrupt present
2	RXRIS	R	Pre-enable receive interrupt flag: 0: Interrupt not present 1: Interrupt present
1	RTRIS	R	Pre-enable timeout interrupt flag: 0: Interrupt not present 1: Interrupt present
0	RORRIS	R	Pre-enable overrun interrupt flag: 0: Interrupt not present 1: Interrupt present

13.3.9 SSPxMIS (Post-enable interrupt status register)

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

Bit	Bit Symbol	Type	Function
31-4	−	W	Write as "0".
3	TXMIS	R	Post-enable transmit interrupt flag: 0: Interrupt not present 1: Interrupt present
2	RXMIS	R	Post-enable receive interrupt flag: 0: Interrupt not present 1: Interrupt present
1	RTMIS	R	Post-enable time-out interrupt flag: 0: Interrupt not present 1: Interrupt present
0	RORMIS	R	Post-enable overrun interrupt flag: 0: Interrupt not present 1: Interrupt present

13.3.10 SSPxICR (Interrupt clear register)

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

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as "0".
1	RTIC	W	Clear the time-out interrupt flag: 0: Invalid 1: Clear
0	RORIC	W	Clear the overrun interrupt flag: 0: Invalid 1: Clear

13.3.11 SSPxDMACR (DMA control register)

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

Bit	Bit Symbol	Type	Function
31-2	−	W	Write as "0".
1	TXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable
0	RXDMAE	R/W	Transmit FIFO DMA control: 0:Disable 1:Enable

13.4 Overview of SSP

This LSI contains the SSP with 2channels.

The SSP is an interface that enables serial communications with the peripheral devices with three types of synchronous serial interface functions.

The SSP performs serial-parallel conversion of the data received from a peripheral device.

The transmit buffers data in the independent 16-bit wide and 8-layered transmit FIFO in the transmit mode, and the receive buffers data in the 16-bit wide and 8-layered receive FIFO in receive mode. Serial data is transmitted via SPxDO and received via SPxDI.

The SSP contains a programmable prescaler to generate the serial output clock SPxCLK from the input clock φT0. The operation mode, frame format, and data size of the SSP are programmed in the control registers SSPxCR0 and SSPxCR1.

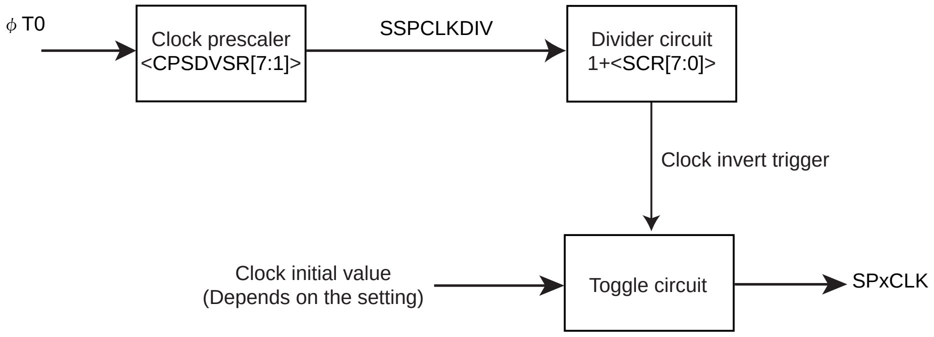
13.4.1 Clock prescaler

When configured as a master, a clock prescaler comprising two free-running serially linked counters is used to provide the serial output clock SPxCLK.

You can program the clock prescaler through the SSPxCPSR register, to divide φT0 by a factor of 2 to 254 in steps of two. Because the least significant bit of the SSPxCPSR register is not used, division by an odd number is not possible.

The output of the prescaler is further divided by a factor of 1 to 256, which is obtained by adding 1 to the value programmed in the SSPxCR0 register, to give the master output clock SPxCLK.

Bitrate = φT0 / (<CPSDVSR>×(1+<SCR>))


ϕ T0
Clock prescaler <CPSDVSR[7:1]>
SSPCLKDIV
Divider circuit 1+<SCR[7:0]>
Clock invert trigger
Clock initial value (Depends on the setting)
Toggle circuit
SPxCLK

13.4.2 Transmit FIFO

This is a 16-bit wide, 8-layered transmit FIFO buffer, which is shared in master and slave modes.

13.4.3 Receive FIFO

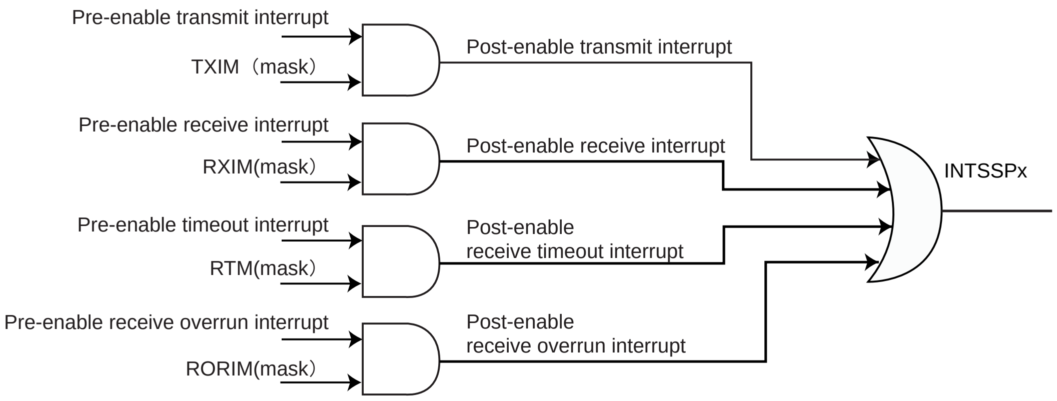
This is a 16-bit wide 8-layered receive FIFO buffer, which is shared in master and slave modes.

13.4.4 Interrupt generation logic

The interrupts, each of which can be masked separately, are generated.

Transmit interrupt	A conditional interrupt to occur when the transmit FIFO has free space more than (including half) of the entire capacity. (Number of valid data items in the transmit FIFO ≤ 4)
Receive interrupt	A conditional interrupt to occur when the receive FIFO has valid data more than half (including half) the entire capacity. (Number of valid data items in the receive FIFO ≥ 4)
Time-out interrupt	A conditional interrupt to indicate that the data exists in the receive FIFO to the time-out period.
Overrun interrupt	Conditional interrupts indicating that data is written to receive FIFO when it is full.

Also, The individual masked sources are combined into a single interrupt. When any of the above interrupts is asserted, the combined interrupt INTSSPx is asserted.


Pre-enable transmit interrupt
TXIM（mask）
Post-enable transmit interrupt
Pre-enable receive interrupt
RXIM(mask)
Post-enable receive interrupt
Pre-enable timeout interrupt
RTM(mask)
Post-enable
receive timeout interrupt
Pre-enable receive overrun interrupt
RORIM(mask)
Post-enable
receive overrun interrupt
INTSSPx

a. Transmit interrupt

The transmit interrupt is asserted when there are four or fewer valid entries in the transmit FIFO. The transmit interrupt is also generated when the SSP operation is disabled (SSPxCR1 <SSE> = "0").

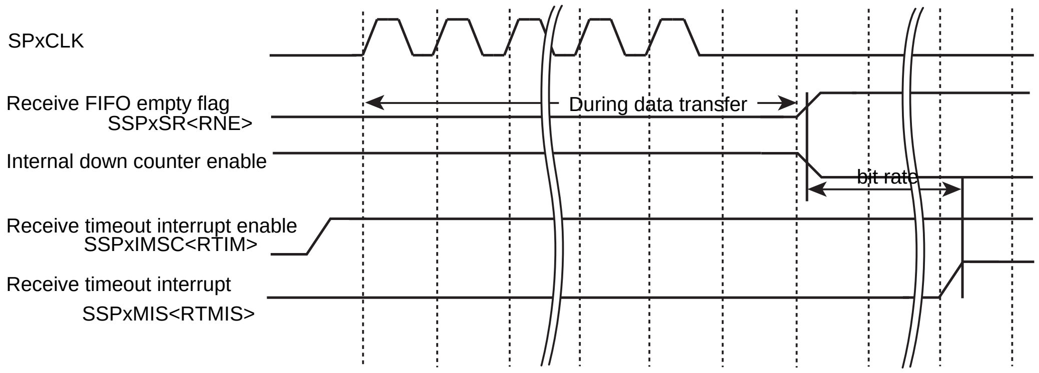
The first transmitted data can be written in the FIFO by using this interrupt.

b. Receive interrupt

The receive interrupt is asserted when there are four or more valid entries in the receive FIFO.

c. Time-out interrupt

The time-out interrupt is asserted when the receive FIFO is not empty and the SSP has remained idle for a fixed 32-bit period (bit rate). This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing. This operation occurs in both master and slave modes. When the time-out interrupt is generated, read all data from the receive FIFO. Even if all the data is not read, data can be transmitted / received if the receive FIFO has a free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. When transfer starts, the timeout interrupt will be cleared. If data is transmitted / received when the receive FIFO has no free space, the time-out interrupt will not be cleared and an overrun interrupt will be generated.


SPxCLK
Receive FIFO empty flag
SSPxSR<RNE>
During data transfer
Internal down counter enable
bit rate
Receive timeout interrupt enable
SSPxIMSC<RTIM>
Receive timeout interrupt
SSPxMIS<RTMIS>

d. Overrun interrupt

When the next data (9th data item) is received when the receive FIFO is already full, an overrun interrupt is generated immediately after transfer. The data received after the overrun interrupt is generated (including the 9th data item) will become invalid and be discarded. However, if data is read from the receive FIFO while the 9th data item is being received (before the interrupt is generated), the 9th received data will be written in the receive FIFO as valid data. To perform transfer properly when the overrun interrupt has been generated, write "1" to SSPxICR<RORIC> register, and then read all data from the receive FIFO. Even if all the data is not read, data can be transmitted / received if the receive FIFO has free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. Note that if the receive FIFO is not read (provided that the receive FIFO is not empty) within a certain 32-bit period (bit rate) after the overrun interrupt is cleared, a time-out interrupt will be generated.

13.4.5 DMA interface

The DMA operation of the SSP is controlled through SSPxDMACR register.

When there are more data than the watermark level (half of the FIFO) in the receive FIFO, the receive DMA request is asserted.

When the amount of data left in the transmit FIFO is less than the watermark level (half of the FIFO), the transmit DMA request is asserted.

To clear the transmit/receive DMA request, an input pin for the transmit/receive DMA request clear signals, which are asserted by the DMA controller, is provided.

Set the DMA burst length to four words.

Note: For the remaining three words, the SSP does not assert the burst request.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described above. All request signals are deasserted if the SSP is disabled or the DMA enable signal is cleared.

The following table shows the trigger points for DMABREQ, for both the transmit and receive FIFOs.

	Burst length	
Watermark level	Transmit (number of empty locations)	Receive (number of filled locations)
1/2	4	4

13.5 SSP operation

13.5.1 Initial setting for SSP

Settings for the SSP communication protocol must be made with the SSP disabled.

Control registers SSPxCR0 and SSPxCR1 need to configure this SSP as a master or slave operating under one of the following protocols. In addition, make the settings related to the communication speed in the clock prescale registers SSPxCPSR and SSPxCR0 <SCR>.

This SSP supports the following protocols:

- SPI
- SSI
- Microwire

13.5.2 Enabling SSP

The transfer operation starts when the operation is enabled with the transmitted data written in the transmit FIFO, or when transmitted data is written in the transmit FIFO with the operation enabled.

However, if the transmit FIFO contains only four or fewer entries when the operation is enabled, a transmit interrupt will be generated. This interrupt can be used to write the initial data.

Note: When the SSP is in the SPI slave mode and the SPxFSS pin is not used, be sure to transmit data of one byte or more in the FIFO before enabling the operation. If the operation is enabled with the transmit FIFO empty, the transfer data will not be output correctly.

13.5.3 Clock ratios

When setting a frequency for φT0, the following conditions must be met.

- In master mode

  f_{SPxCLK} (maximum) → φT0 /2

  f_{SPxCLK} (minimum) → φT0/(254×256)

- In slave mode

  f_{SPxCLK} (maximum) → φT0 /12

  f_{SPxCLK} (minimum) → φT0 /(254×256)

Note: The maximum baud-rate in the master mode is equal or less than 10Mbps.

13.6 Frame Format

Each frame format is between 4 and 16 bits wide depending on the size of data programmed, and is transmitted starting from the MSB.

- Serial clock (SPxCLK)

  Signals remain "Low" in the SSI and Microwire formats and as inactive in the SPI format while the SSP is in the idle state. In addition, data is output at the set bit rate only during data transmission.

- Serial frame (SPxFSS)

  In the SPI and Microwire frame formats, signals are set to "Low" active and always asserted to "Low" during frame transmission.

  In the SSI frame format, signals are asserted only during 1 bit rate before each frame transmission. In this frame format, output data is transmitted at the rising edge of SPxCLK and the input data is received at its falling edge.

Refer to Section "13.6.1" to "13.6.3" for details of each frame format.

13.6.1 SSI frame format

In this mode, the SSP is in idle state, SPxCLK and SPxFSS are forcedly set to "Low", and the transmit data line SPxDO becomes Hi-Z. When data is written in the transmit FIFO, the master outputs "High" pulses of 1 SPxCLK to the SPxFSS line. The transmitted data will be transferred from the transmit FIFO to the transmit serial shift register. Data of 4 to 16 bits will be output from the SPxDO pin at the next rising edge of SPxCLK.

Likewise, the received data will be input starting from the MSB to the SPxDI pin at the falling edge of SPxCLK. The received data will be transferred from the serial shift register into the receive FIFO at the rising edge of SPxCLK after its LSB data is latched.

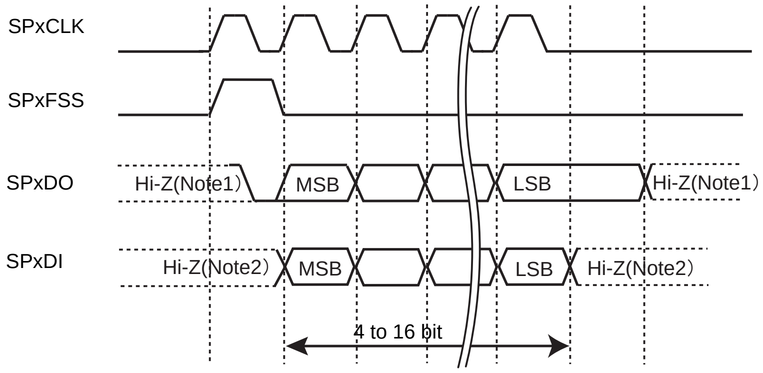


Figure 13-2 SSI frame format (transmission/reception during single transfer)

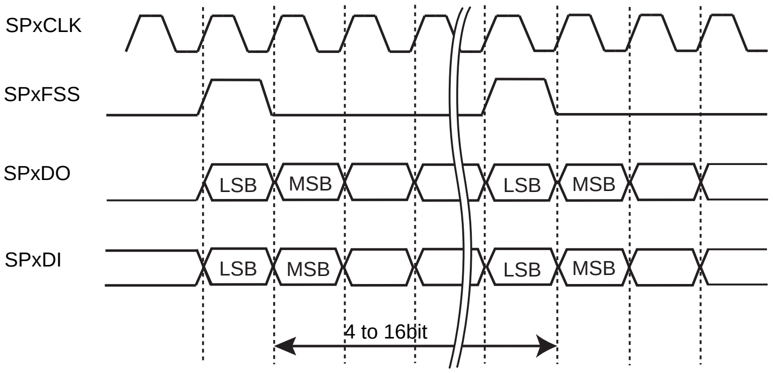


Figure 13-3 SSI frame format (transmission/reception during continuous transfer)

Note 1: When transmission is disable, SPxDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note 2: SPxDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

13.6.2 SPI frame format

The SPI interface has 4 lines. SPxFSS is used for slave selection. One of the main features of the SPI format is that the <SPO> and <SPH> bits in the SSPxCR0 register can be used to set the SPxCLK operation timing.

SSPxCR0 <SPO> is used to set the level at which SPxCLK in idle state is held.

SSPxCR0 <SPH> is used to select the clock edge at which data is latched.

	SSPxCR0<SPO>	SSPxCR0<SPH>
0	"Low" state	Capture data at the 1st clock edge.
1	"High" state	Capture data at the 2nd clock edge.

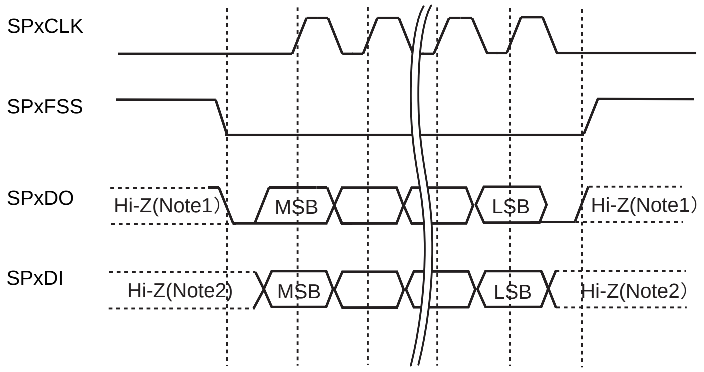


Figure 13-4 SPI frame format (single transfer, <SPO>="0" & <SPH>="0")

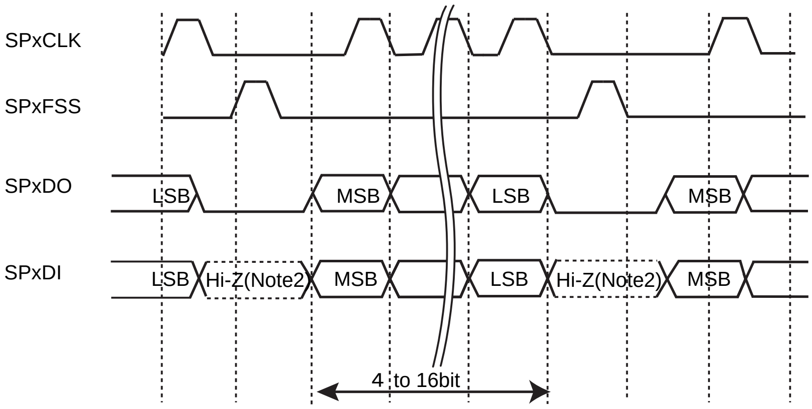


Figure 13-5 SPI frame format (continuous transfer,<SPO>="0" & <SPH>="0")

Note 1: When transmission is disable, SPxDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note 2: SPxDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

With this setting <SPO>="0", during the idle period:

- The SPxCLK signal is set to "Low".
- SPxFSS is set to "High".
- The transmit data line SPxDO is set to "Low".

If the SSP is enabled and valid data exists in the transmit FIFO, the SPxFSS master signal driven by "Low" notifies of the start of transmission. This enables the slave data in the SPxDI input line of the master.

When a half of the SPxCLK period has passed, valid master data is transferred to the SPxDO pin. Both the master data and slave data are now set. When another half of SPxCLK has passed, the SPxCLK master clock pin becomes "High". After that, the data is captured at the rising edge of the SPxCLK signal and transmitted at its falling edge.

In the single transfer, the SPxFSS line will return to the idle "High" state when all the bits of that data word have been transferred, and then one cycle of SPxCLK has passed after the last bit was captured.

However, for continuous transfer, the SPxFSS signal must be pulsed at HIGH between individual data word transfers. This is because change is not enabled when the slave selection pin freezes data in its peripheral register and the <SPH> bit is logical 0.

Therefore, to enable writing of serial peripheral data, the master device must drive the SPxFSS pin of the slave device between individual data transfers. When the continuous transfer is completed, the SPxFSS pin will return to the idle state when one cycle of SPxCLK has passed after the last bit is captured.

13.6.3 Microwire frame format

The Microwire format uses a special master/slave messaging method, which operates in half-duplex mode. In this mode, when a frame begins, an 8-bit control message is transmitted to the slave. During this transmission, no incoming data is received by the SSP. After the message has been transmitted, the slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, it responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

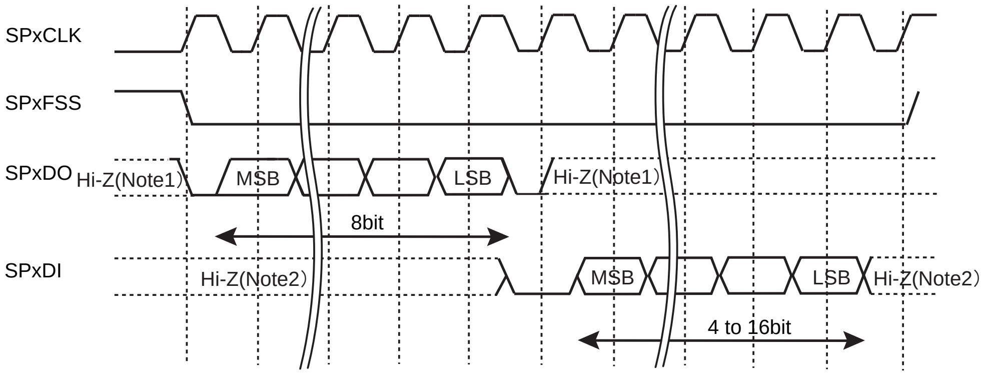


Figure 13-6 Microwire frame format (single transfer)

Note 1: When transmission is disabled, SPxDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Note 2: SPxDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Though the Microwire format is similar to the SPI format, it uses the master/slave message transmission method for half-duplex communications. Each serial transmission is started by an 8-bit control word, which is sent to the off-chip slave device. During this transmission, the SSP does not receive input data. After the message has been transmitted, the off-chip slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits. With this configuration, during the idle period:

- The SPxCLK signal is set to "Low".
- SPxFSS is set to "High".
- The transmit data line SPxDO is set to "Low".

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SPxFSS causes the value stored in the bottom entry of the transmit FIFO to be transferred to the serial shift register for the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SPxDO pin.

SPxFSS remains "Low" and the SPxDI pin remains tristated during this transmission. The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SPxCLK.

After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSP. Each bit is driven onto SPxDI line on the falling edge of SPxCLK.

The SSP in turn latches each bit on the rising edge of SPxCLK. At the end of the frame, for single transfers, the SPxFSS signal is pulled "High" one clock period after the last bit has been latched in the receive serial shifter, which causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SPxCLK after the LSB has been latched by the receive shifter, or when the SPxFSS pin goes "High".

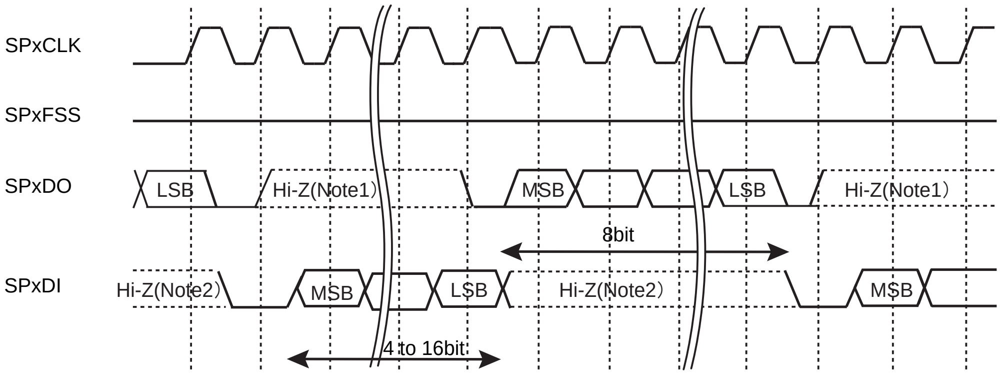


Figure 13-7 Microwire frame format (continuous transfer)

Note 1: When transmission is disabled, SPxDO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to fix the voltage level.

Note 2: SPxDI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to fix the voltage level.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SPxFSS line is continuously asserted (held Low) and transmission of data occurs back to back.

The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge of SPxCLK, after the LSB of the frame has been latched into the SSP.

Note: [Example of connection] The SSP does not support dynamic switching between the master and slave in the system. Each sample SSP is configured and connected as either a master or slave.

14. Remote control signal preprocessor(RMC)

14.1 Basic operation

Remote control signal preprocessor (hereafter referred to as RMC) receives a remote control signal of which carrier is removed.

14.1.1 Reception of Remote Control Signal

- A sampling clock can be selected from either low frequency clock (32.768kHz) or Timer output.
- Noise canceling time can be adjusted.
- Leader detection
- Batch reception up to 72bit of data

14.2 Block Diagram

Figure 14-1 shows the block diagram of RMC.

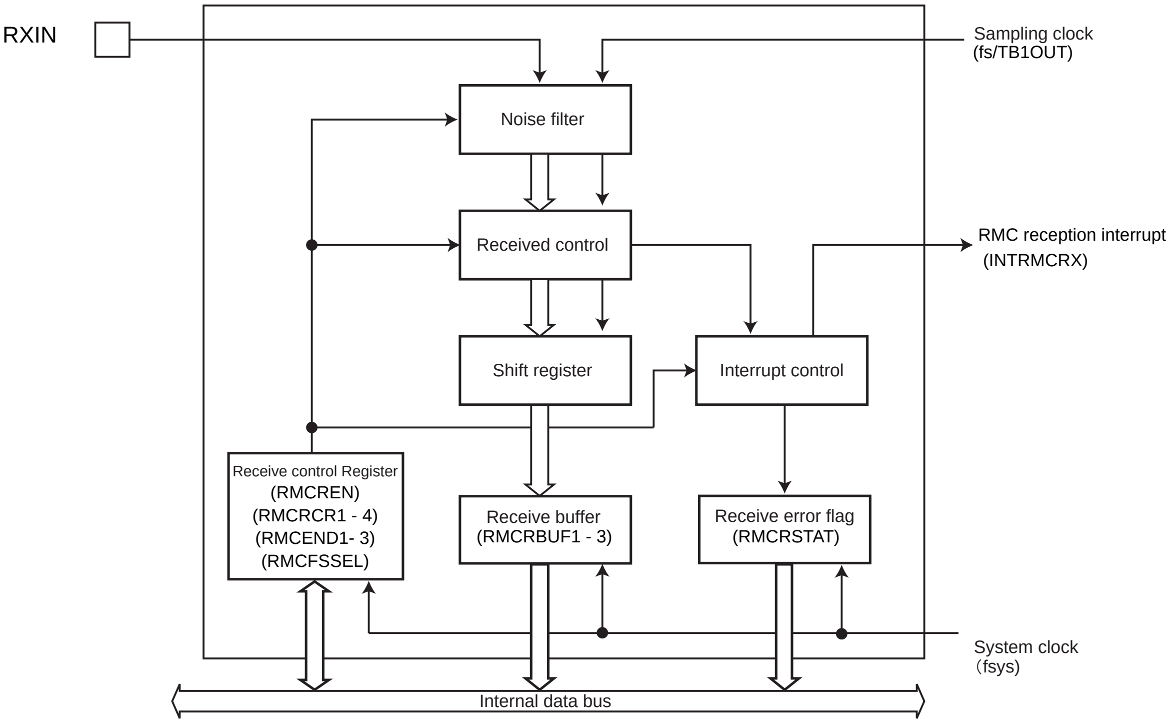


Figure 14-1 Block diagram of RMC

14.3 Registers

14.3.1 Register List

Addresses and names of RMC control registers are shown below.

Base Address = 0x4004_0400

Register		Address(Base+)
Enable Register	RMCEN	0x0000
Receive Enable Register	RMCREN	0x0004
Receive Data Buffer Register 1	RMCRBUF1	0x0008
Receive Data Buffer Register 2	RMCRBUF2	0x000C
Receive Data Buffer Register 3	RMCRBUF3	0x0010
Receive Control Register 1	RMCRCR1	0x0014
Receive Control Register 2	RMCRCR2	0x0018
Receive Control Register 3	RMCRCR3	0x001C
Receive Control Register 4	RMCRCR4	0x0020
Receive Status Register	RMCRSTAT	0x0024
Receive End bit Number Register 1	RMCEND1	0x0028
Receive End bit Number Register 2	RMCEND2	0x002C
Receive End bit Number Register 3	RMCEND3	0x0030
Source Clock selection Register	RMCFSSEL	0x0034

14.3.2 RMCEN(Enable Register)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	−	R/W	Write as "1".
0	RMCEN	R/W	Controls RMC operation. 0: Disabled 1: Enabled To allow RMC to function, enable the RMCEN bit first. If the operation is disabled, all the clocks for RMC except for the enable register are stopped, and it can reduce power consumption. If RMC is enabled and then disabled, the settings in each register remain intact.

14.3.3 RMCREN(Receive Enable Register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	RMCREN	R/W	Reception 0: Disabled 1: Enabled Controls reception of RMC. Setting this bit to "1" enables reception.

Note: Enable the <RMCREN> bit after setting the RMCRCR1, RMCRCR2, and RMCRCR3.

14.3.4 RMCRBUF1(Receive Data Buffer Register 1)

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

Bit	Bit Symbol	Type	Function
31-0	RMCRBUF[31:0]	R	Received data (31 to 0 bit) Reads 4 bytes of received data. (31 to 0 bit)

14.3.5 RMCRBUF2(Receive Data Buffer Register 2)

	31	30	29	28	27	26	25	24
bit symbol	RMCRBUF(Received data 63 to 54 bit)							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RMCRBUF(Received data 55 to 48 bit)							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMCRBUF(Received data 47 to 40 bit)							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCRBUF(Received data 39 to 32 bit)							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	RMCRBUF[63:32]	R	Received data (63 to 32 bit) Reads 4 bytes of received data. (63 to 32 bit)

14.3.6 RMCRBUF3(Receive Data Buffer Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMCRBUF(Received data 71 to 64 bit)							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	RMCRBUF[71:64]	R	Received data (71 to 64 bit). Reads 1 byte of received data. (71 to 64 bit).

Note: The received bit is stored in the data buffer register in MSB-first order, and the last received bit is stored in the LSB (bit 0). If the remote control signal is received in the LSB first algorithm, the received data is stored in reverse sequence.

14.3.7 RMCRCR1(Receive Control Register 1)

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

Bit	Bit Symbol	Type	Function
31-24	RMCLCMAX[7:0]	R/W	Specifies a maximum cycle of leader detection. Calculating formula of the maximum cycle: <RMCLCMAX> × 4/fs [s].
23-16	RMCLCMIN[7:0]	R/W	Specifies a minimum cycle of leader detection. Calculating formula of the minimum cycle: <RMCLCMIN> × 4/fs [s].
15-8	RMCLLMAX[7:0]	R/W	Specifies a maximum low width of leader detection. Calculating formula of the maximum low width: <RMCLLMAX> × 4/fs [s]
7-0	RMCLLMIN[7:0]	R/W	Specifies a minimum low width of leader detection. Calculating formula for the minimum low width: <RMCLLMIN> × 4/fs [s] When RMCRCR2<RMCLD> = 1, a value of the low-pulse width is less than the specified value, it is defined as data bit.

Note: When you configure the register, you must follow the rule shown below.

Leader	Rules
Low width + High width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> > <RMCLLMIN[7:0]> <RMCLCMIN[7:0]> > <RMCLLMAX[7:0]>
Only high width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> = 0x00 <RMCLLMIN[7:0]> = don't care
No Leader	<RMCLCMAX[7:0]> = 0x00 <RMCLCMIN[7:0]> = don't care <RMCLLMAX[7:0]> = don't care <RMCLLMIN[[7:0]> = don't care

14.3.8 RMCRCR2(Receive Control Register 2)

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

Bit	Bit Symbol	Type	Function
31	RMCLIEN	R/W	Leader detection interrupt 0: Not generated 1: Generated
30	RMCEDIEN	R/W	Remote control input falling edge interrupt 0: Not generated 1: Generated
29-26	−	R	Read as 0.
25	RMCLD	R/W	Receiving remote control signal with or without leader 0: Disabled 1: Enabled
24	RMCPHM	R/W	Receiving a remote control signal by a phase modulation 0: Not receiving a remote control signal by a phase modulation. (receive by a cycle modulation) 1: Receive remote control signal by a fixed-frequency pulse modulation. To receive a fixed-frequency remote control signal by a pulse modulation, set this bit to "1".
23-16	−	R	Read as 0.
15-8	RMCLL[7:0]	R/W	Excess low width that triggers reception completion and interrupt generation. 0000_0000 to 1111_1110: Reception completion and interrupt generation at <RMCLL> × 1/fs [s]. 1111_1111: not to use as the trigger
7-0	RMCDMAX[7:0]	R/W	Maximum data bit cycle that triggers reception completion and interrupt generation. 0000_0000 to 1111_1110: Reception completion and interrupt generation at <RMCDMAX> × 1/fs [s]. 1111_1111: not to use as the trigger

14.3.9 RMCRCR3(Receive Control Register 3)

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

Bit	Bit Symbol	Type	Function
31-15	-	R	Read as 0.
14-8	RMCDATH[6:0]	R/W	Larger threshold to determine a signal pattern in a phase method Calculating formula of the threshold: <RMCDATH> × 1/fs [s] Specifies a larger threshold (within a range of 1.5T and 2T) to determine a pattern of remote control signal in a phase method. If the measured cycle exceeds the threshold, the bit is determined as "10". If not, the bit is determined as "01".
7	-	R	Read as 0.
6-0	RMCDATL[6:0]	R/W	Threshold to determine 0 or 1 smaller threshold to determine a signal pattern in a phase method. Calculating formula of the threshold: <RMCDATL> × 1/fs [s] Specifies two kinds of thresholds: a threshold to determine whether a data bit is 0 or 1; a smaller threshold (within a range of 1T and 1.5T) to determine a pattern of remote control signal in a phase method. As for the determination of data bit, if the measured cycle exceeds the threshold, the bit is determined as "1". If not, the bit is determined as "0". Calculating formula of the threshold: <RMCDATL> × 1/fs [s]. As for the determination of a remote control signal pattern in a phase method, if the measured cycle exceeds the threshold, the bit is determined as "01". If not, the bit is determined as "00".

Note: If the <RMCPHM> bit of the Receive Control Register 2 is "0", <RMCDATH[6:0]> are not enabled.
The bits are enabled when <RMCPHM> is "1".

14.3.10 RMCRCR4(Receive Control Register 4)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	RMCPO	R/W	Remote control input signal 0: Not reversed 1: Reversed
6-4	−	R	Read as 0.
3-0	RMCNC[3:0]	R/W	Specifies noise cancellation time. 0000: No cancellation 0001 to 1111: cancellation Calculating formula of noise cancellation time: <RMCNC> × 1/fs [s]

14.3.11 RMCRSTAT(Receive Status Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15	RMCRLIF	R	Interrupt source flag 0: No leader detection interrupt generated. 1: Leader detection interrupt generated.
14	RMCLOIF	R	Interrupt source flag 0: No low width detection interrupt generated. 1: Low width detection interrupt generated.
13	RMCDMAXIF	R	Interrupt source flag 0: No maximum data bit cycle interrupt generated. 1: Maximum data bit cycle interrupt generated.
12	RMCEDIF	R	Interrupt source flag 0: No falling edge interrupt generated. 1: Falling edge interrupt generated.
11-8	−	R	Read as 0.
7	RMCRLDR	R	Leader detection. 0: Disable leader detection. 1: Enable leader detection.
6-0	RMCRNUM[6:0]	R	The number of received data bit 000_0000:no data bit (only with leader) 000_0001 to 100_1000: 1 to 72bit 100_1001 to 111_1111: 73bit and more Indicates the number of bits received as remote control signal data. The number cannot be monitored during reception. On completion of reception, the number is stored.

Note 1: This register is updated every time an interrupt is generated. Writing to this register is ignored.

Note 2: RMC keeps receiving 73 bit or more data unless reception is completed by detecting the maximum data bit cycle or the excess low width. In this case, the received data in the data buffer may not be ensured.

14.3.12 RMCEND1(Receive End bit Number Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	RMCEND1						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6-0	RMCEND1[6:0]	R/W	Specifies that the number of receive data bit 000_0000 : No specifically the receive data bit 000_0001 to 100_1000 : Specifies that the number of receive data bit(1 to 72bit) 100_1001 to 111_1111 : Don’t set the value

14.3.13 RMCEND2(Receive End bit Number Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	RMCEND2						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6-0	RMCEND2[6:0]	R/W	Specifies that the number of receive data bit 000_0000 : No specifically the receive data bit 000_0001 to 100_1000 : Specifies that the number of receive data bit(1 to 72bit) 100_1001 to 111_1111 : Don’t set the value

14.3.14 RMCEND3(Receive End bit Number Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	RMCEND3						
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6-0	RMCEND3[6:0]	R/W	Specifies the number of receive data bit 000_0000 : No specifically the receive data bit 000_0001 to 100_1000 : Specifies that the number of receive data bit(1 to 72bit) 100_1001 to 111_1111 : Don’t set the value

Note 1: As specified to RMCEND1, RMCEND2 and RMCEND3, it is able to set three kinds of the receive data bit.

Note 2: To use the RMCEND1, RMCEND2 and RMCEND3 is in combination with the maximum data bit cycle.

14.3.15 RMCFSSEL(Source Clock selection Register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	RMCCLK	R/W	Specifies that Sampling clock of RMC function 0 : Low frequency Clock (32.768kHz) 1 : Timer output(TB1OUT) For the Sampling of RMC function, It is able to set the Low Frequency Clock (32.768kHz) or Timer output (TB1OUT). The Setting range of Timer output by TB1OUT is from 30 to 34kHz.

Note: To Change the sampling clock by using the RMCFSSEL, disable the RMC operation first by using the RMCEN<RMCEN>. Then, enable it again, and set the RMCFSSEL before setting other RMC registers.

14.4 Operation Description

14.4.1 Reception of Remote Control Signal

14.4.1.1 Sampling clock

A remote control signal is sampled by using low-speed 32.768kHz clock (fs).

14.4.1.2 Basic operation

RMC set RMCRSTAT<RMCRLDR> bit when a leader is detected.

At this time, if you set the RMCRCR2<RMCLIEN> bit, leader detection will generate a leader detection interrupt. When a leader detection interrupt occurs, RMCRSTAT<RMCRLIF> bit is set.

After the leader detecting, each data bit is determined as "0" or "1" in sequence. The results are stored in RMCRBUF1, RMCRBUF2 and RMCRBUF3 registers up to 72 bits. By setting RMCRCR2< RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of data bit. When a remote control signal input falling edge interrupt is generated, RMCRSTAT< RMCEDIF > bit is set.

Data reception stops when the maximum data bit cycle is detected and low-width matches the setting value, and then, an interrupt occurs. If <RMCEND1>, <RMCEND2> and <RMCEND3> of the register RMCxEND1, RMCxEND2 and RMCEND3 have been configured, data reception stops and an interrupt occurs only in the case that the number of bits received before maximum data bit cycle is detected. The condition of RMC can be checked by reading the remote control receive status register.

To check the status of RMC if reception is completed, read the remote control receive status register.

On completion of reception, RMC is waiting for the next leader.

By setting RMC to receive a signal without a leader, RMC recognizes the received as data and starts reception without detecting a leader.

If the next data reception is completed before reading the preceding received data, the preceding data is overwritten by the next one.

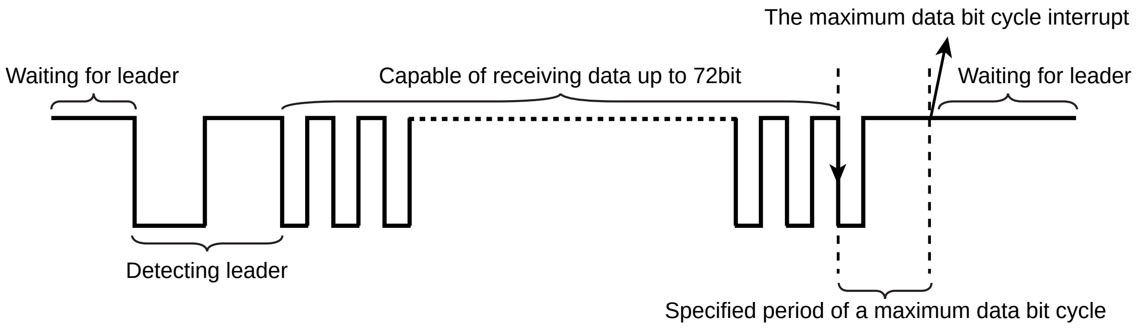

The maximum data bit cycle interrupt
Waiting for leader
Capable of receiving data up to 72bit
Waiting for leader
Detecting leader
Specified period of a maximum data bit cycle

Figure 14-2 Data reception completed by detecting the max data bit cycle

14.4.1.3 Preparation

Before starting receiving process, configure how to receive remote control signal using the Remote Control Signal Receive Control Registers (RMCRCR1, RMCRCR2 and RMCRCR3, RMCRCR4).

(1) Settings of Noise Cancelling Time

Configure noise cancelling time with the RMCRCR4 <RMCNC[3:0]> bit.

Noise canceling is applied to remote control signals sampled by the sampling clock.

RMC monitors a sampled remote control signal in each rising edge of a sampling clock. If "High" is monitored, RMC recognizes that the signal was changed to "Low" after monitoring cycles of "Low"s specified in <RMCNC>. If "Low" is monitored, RMC recognizes that the signal was changed to "High" after monitoring cycles of "High" specified in <RMCNC>.

The following figure shows how RMC operates according to the noise cancel setting of <RMCNC [3:0]> = "0011" (3 cycles). Subsequent to noise cancellation, the signal is changed from "High" to "Low" upon monitoring 3 cycles of "Low", and the signal is changed from "Low" to "High" upon monitoring 3 cycles of "High".

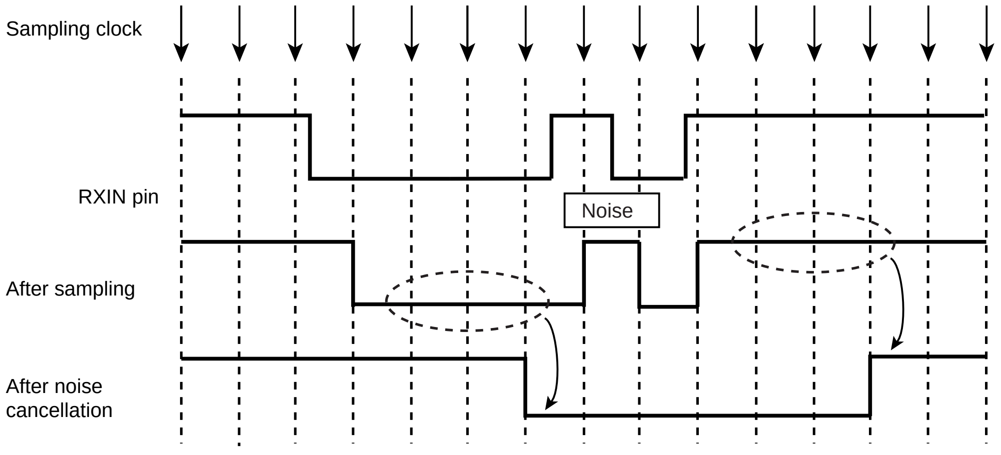


Figure 14-3 Noise Cancel (In the case of RMCRCR4="0011" (3 Cycles))

(2) Settings of Detecting Leader

Set the leader cycle and a low width of the leader to RMCRCR1 <RMCLLMIN[7:0]> <RMCLLMAX[7:0]> <RMCLCMIN[7:0]> <RMCLCMAX[7:0]> bits. When you configure those above, follow the rule shown below.

Leader	Rules
Low width + High Width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> > <RMCLLMIN[7:0]> <RMCLCMIN[7:0]> > <RMCLLMAX[7:0]>
Only high width	<RMCLCMAX[7:0]> > <RMCLCMIN[7:0]> <RMCLLMAX[7:0]> = 0000_0000 <RMCLLMIN[7:0]> = don't care
No leader	<RMCLCMAX[7:0]> = 0000_0000 <RMCLCMIN[7:0]> = don't care <RMCLLMAX[7:0]> = don't care <RMCLLMIN[7:0]> = don't care

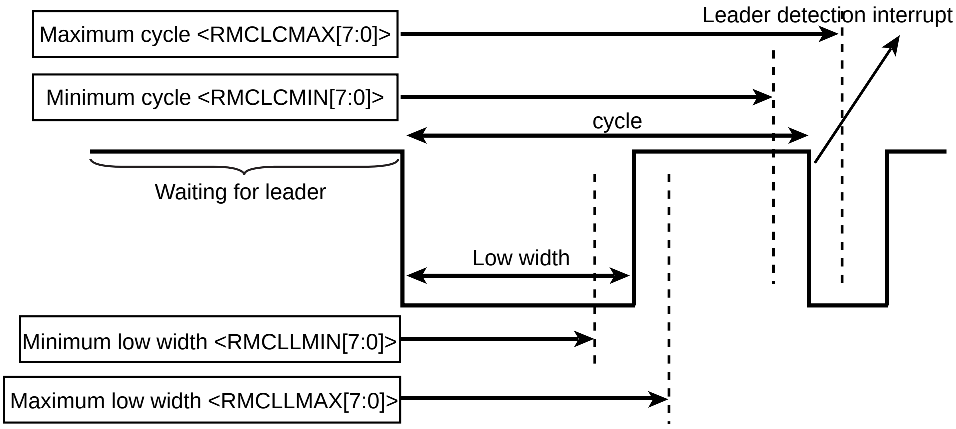

Leader detection interrupt
Maximum cycle <RMCLCMAX[7:0]>
Minimum cycle <RMCLCMIN[7:0]>
cycle
Waiting for leader
Low width
Minimum low width <RMCLLMIN[7:0]>
Maximum low width <RMCLLMAX[7:0]>

Figure 14-4 Leader wave form and the RMCRCR1 register settings

If you want to generate an interrupt when detecting a leader, configure the RMCRCR2 <RMCLIEN> bit.

A remote control signal without a leader cannot generate a leader detection interrupt.

(3) Setting of 0/1 determination data bit

Based on a falling edge cycle, the data bit of a cycle modulation is determined as 0 or 1.

There are two kinds of determinations:

As for data bit determination of a remote control signal in a phase method, see"14.4.1.8 Receiving a Remote Control Signal in a Phase Method".

1. Determination by threshold.

   Configure a threshold value to RMCRCR3<RMCDATL[6:0]> bit which determines data bit as "0" or "1." If the determination value is equal to threshold value or more, it is determined as "1." If the determination value is less than threshold value, it is determined as "0."

2. Determination by falling edge interrupt inputs.

   By setting "1" to the RMCRCR2<RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of the data bit. Using this interrupt together with a timer enables the determination to be done by software.

The followings shows the determination method of data bit.

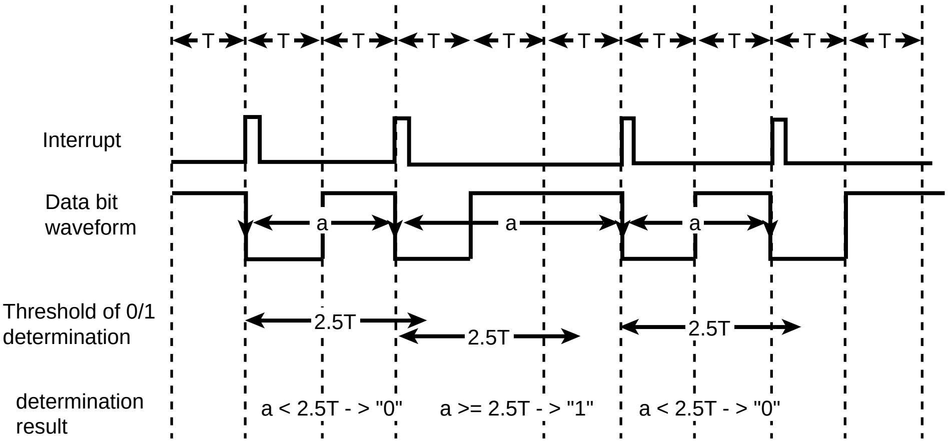


Figure 14-5 Determination method of data bit (In case that threshold is 2.5T)

(4) Settings of Reception Completion

To complete data reception, settings of detecting the maximum data bit cycle and excess low width are required. If multiple factors are specified, reception is completed by the factor detected first. Make sure to configure the reception completion settings.

1. completion by the maximum data bit cycle

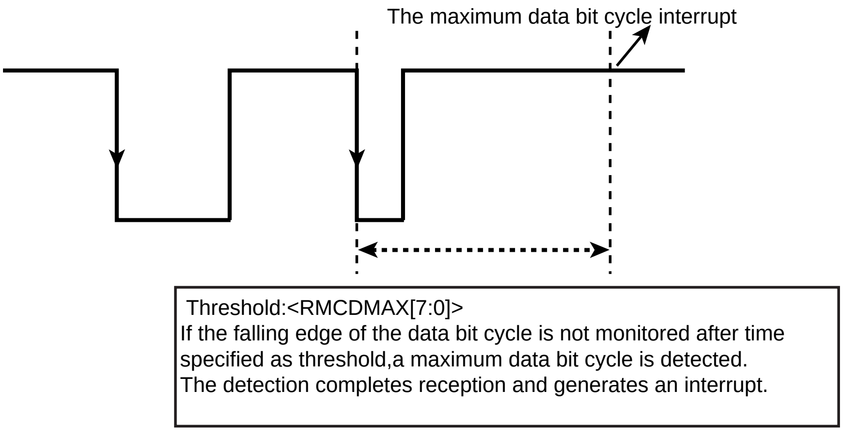
   To complete reception by detecting a maximum data bit cycle, you need to configure the RMCRCR2 <RMCDMAX[7:0]> bits.

   If the falling edge of the data bit cycle isn't monitored after time specified as threshold in the <RMCDMAX[7:0]> bits, a maximum data bit cycle is detected. The detection completes reception and generates an interrupt. After interrupt inputs generated, RMCRSTAT< RMCDMAXIF > bit is set to "1".

   To complete reception by setting the number of receive data is set a RMCEND 1 to 3 register of each <RMCEND1>, <RMCEND2>, <RMCEND3>.In this case when the number of set reception bit agreed with the number of bit which received at the time of the outbreak of MAX on the number of receive data is set a RMCEND 1 to 3 register of each <RMCEND1>, <RMCEND2>, <RMCEND3>, it occurs by an MAX interrupt in data bit period.

   As specified to RMCEND1 to 3, it is able to set three kinds of the receive data bit.

   When it can receive the Maximum Data bit , the number of bit is not match the setting value in <RMCEND1>, <RMCEND2>, <RMCEND3>, it wait for Leader Reception.



Threshold:<RMCDMAX[7:0]>
If the falling edge of the data bit cycle is not monitored after time specified as threshold,a maximum data bit cycle is detected.
The detection completes reception and generates an interrupt.

Figure 14-6 completion by the maximum data bit cycle

2. Completion by detecting low width

   To complete reception by detecting the low width, you need to configure the RMCRCR2 <RMCLL[7:0]> bits.

   After the falling edge of the data bit is detected, if the signal stays low longer than specified, excess low width is detected. The detection completes reception and generates an interrupt.

   After interrupt inputs generated, RMCRSTAT<RMCLOIF> bit is set to "1."

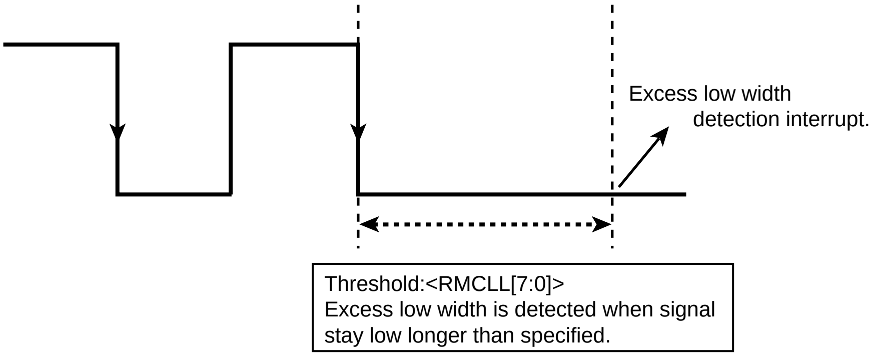

Excess low width
detection interrupt.
Threshold:<RMCLL[7:0]>
Excess low width is detected when signal
stay low longer than specified.

Figure 14-7 Completion by detecting low width

14.4.1.4 Enabling Reception

By enabling the RMCREN <RMCREN> bit after configuring the RMCRCR1, RMCRCR2, RMCRCR3 and RMCRCR4 registers, RMC is ready for reception. Detecting a leader initiates reception.

Note: Changing the configurations of the RMCRCR1, RMCRCR2, RMCRCR3, RMCRCR4, RMCEND1, RMCEND2 or RMCEND3 registers during reception may harm their proper operation. Be careful if you change them during reception.

14.4.1.5 Stopping Reception

RMC stops reception by clearing the RMCREN <RMCREN> bit to "0" (reception disabled).

Clearing this bit during reception stops reception immediately and the received data is discarded.

14.4.1.6 Receiving Remote Control Signal without Leader in Waiting Leader

Setting RMCRCR2 <RMCLD> enables RMC to receive signals with or without a leader.

By setting RMCRCR2 <RMCLD>, RMC starts receiving data if it recognizes a signal of which low width is shorter than a maximum low width of leader detection specified in the RMCRCR1 <RMCLLMAX [7:0]> bits. RMC keeps receiving data until the final data bit is received.

If RMCRCR2 <RMCLD> is enabled, the same settings of error detection, reception completion and data bit determination of 0 or 1 are applied regardless of whether a signal has a leader or not.

Thus receivable remote control signals are limited.

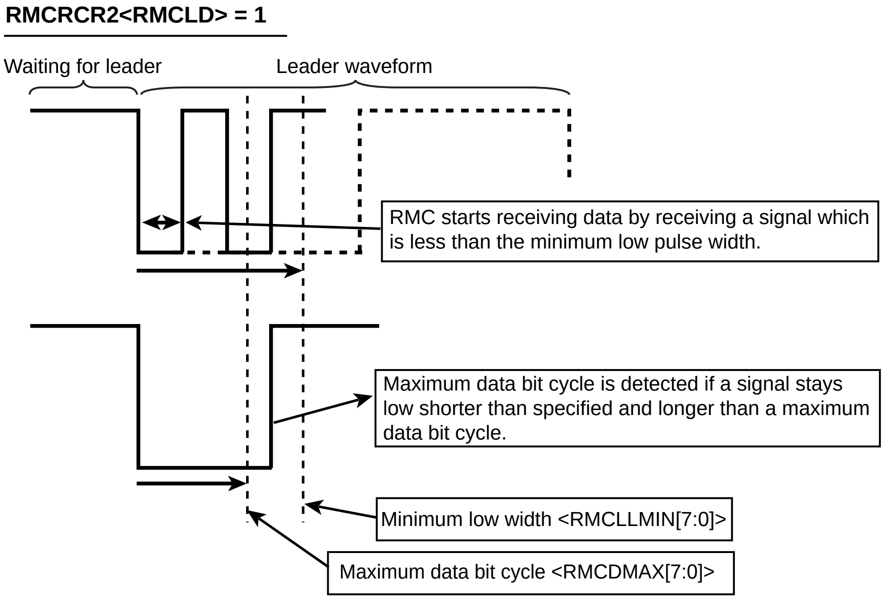

RMCRCR2<RMCLD> = 1
Waiting for leader
Leader waveform
RMC starts receiving data by receiving a signal which is less than the minimum low pulse width.
Maximum data bit cycle is detected if a signal stays low shorter than specified and longer than a maximum data bit cycle.
Minimum low width <RMCLLMIN[7:0]>
Maximum data bit cycle <RMCDMAX[7:0]>

Figure 14-8 Receiving Remote ControlRCR2<RMCLD>="1")

14.4.1.7 A Leader only with Low Width

The figure shown below illustrates a remote control signal that starts with a leader of which waveform only has low width.

This signal starts with a leader that only has low width and a data bit cycle starts from the rising edge. To enable the signal, it must be sent after being reversed by setting the RMCRCR4 <RMCPO> bit to "1".

This is because RMC is configured to detect a data bit cycle from the falling edge

To detect a leader, configure only low-pulse width of the leader with the <RMCLLMAX[7:0]> = "0000_0000", <RMCLCMAX[7:0]> > <RMCLCMIN[7:0]>.

In this case, the value of <RMCLLMIN[7:0]> is set as "don't care".

To detect whether data "0" or data "1", configure the threshold of 0/1 detection with the RMCRCR3 <RMCDATL[6:0]>.

The maximum data bit cycle is configured with the <RMCDMAX[7:0]> of the RMCRCR2.

To complete data reception, configure the maximum data bit cycle with <RMCDMAX[7:0]> of the RMCRCR2, and configure the low-pulse width detection with <RMCLL[7:0]>.

After detecting the maximum data bit cycle and confirming the low-pulse with which is specified after receiving the last bit, receiving data is completed.

The RMC generates an interrupt and waits for the next leader.

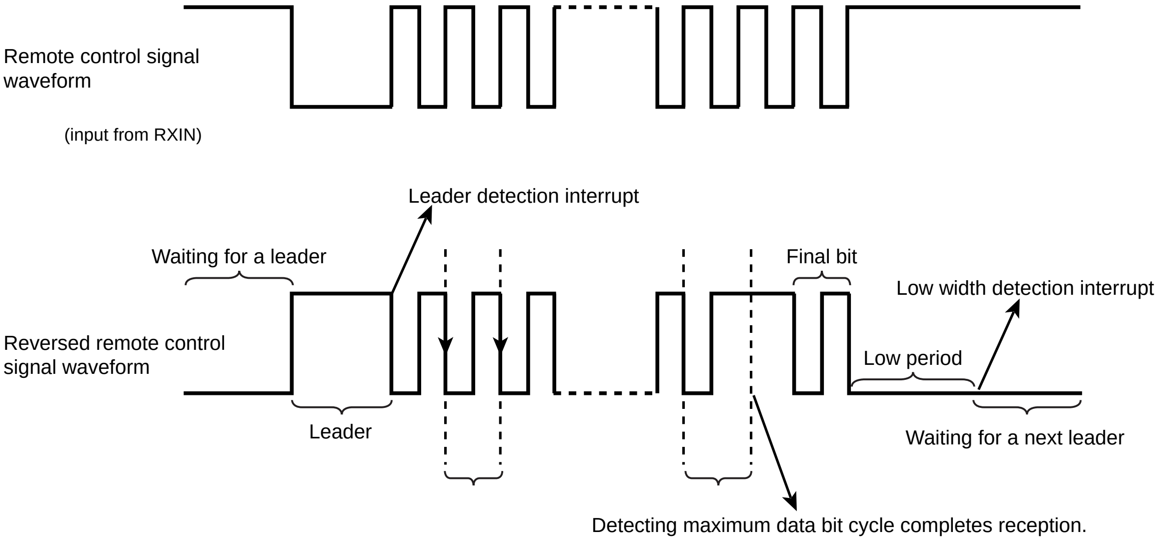

Remote control signal waveform
(input from RXIN)
Leader detection interrupt
Waiting for a leader
Final bit
Low width detection interrupt
Reversed remote control signal waveform
Low period
Leader
Waiting for a next leader
Detecting maximum data bit cycle completes reception.

Figure 14-9 A Leader only with Low Width

14.4.1.8 Receiving a Remote Control Signal in a Phase Method

RMC is capable of receiving a remote control signal in a phase method of which signal cycle is fixed. A signal in the phase method has three waveform patterns (see the figure shown below).

By setting two thresholds a remote control signal pattern is determined. RMC converts the signal into data "0" or "1". On completion of reception, received data "0" and "1" are stored in the RMCRBUF1, RMCRBUF 2 and RMCRBUF3.

By setting RMCRCR2<RMCPHM> = "1", RMC enables to receive a remote control signal in the phase method. Each threshold can be configured with the RMCRCR3 <RMCDATL[6:0]> bits and <RMCDATH[6:0]> bits.

Two thresholds are used to distinguish three waveform patterns. On condition that a cycle between two falling edges is "T", three patterns show cycles of 1T, 1.5T and 2T. Details of the two thresholds are shown below.

	Determined by	Threshold	Register bits to set
Threshold 1	Pattern 1 & pattern 2	1T to 1.5T	RMCRCR3<RMCDATL[6:0]>
Threshold 2	Pattern 2 & pattern 3	1.5T to 2T	RMCRCR3<RMCDATH[6:0]>

To determine a remote control signal in the phase method, three patterns of data waveform and preceding data are required. In addition, the signal needs to start from data "11".

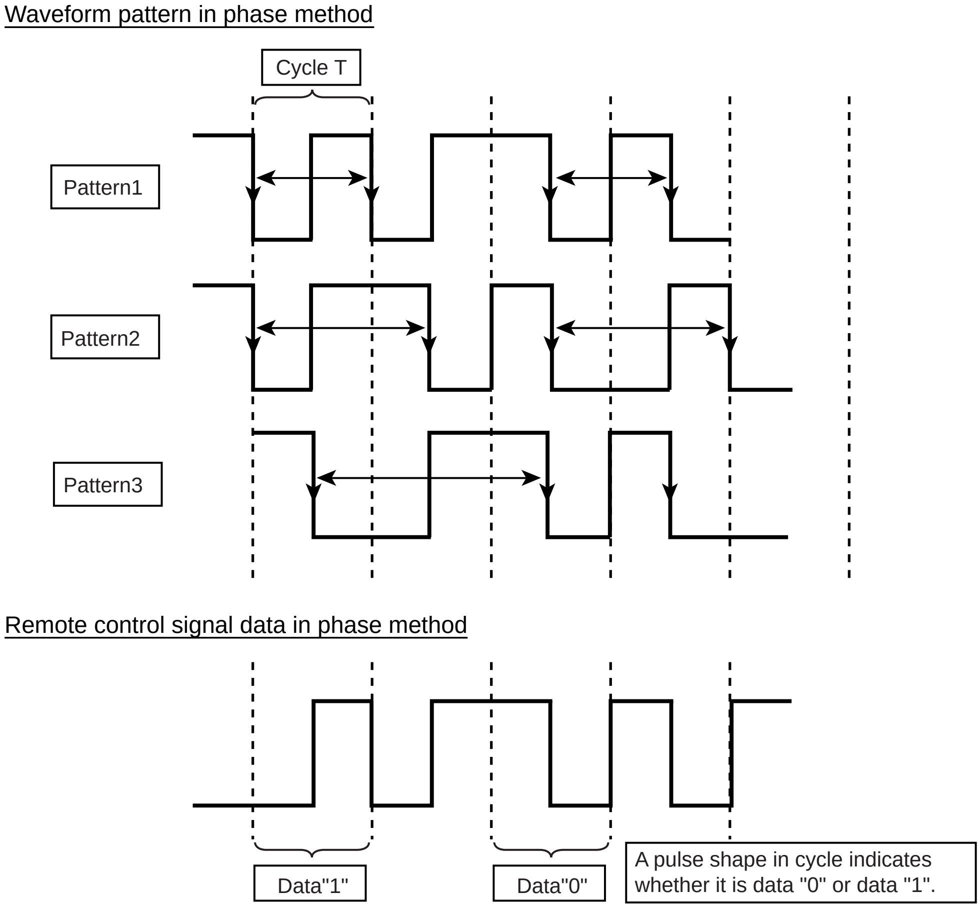

Waveform pattern in phase method
Cycle T
Pattern1
Pattern2
Pattern3
Remote control signal data in phase method
Data"1"
Data"0"
A pulse shape in cycle indicates whether it is data "0" or data "1".

Figure 14-10 Waveform pattern in phase method and the example of data

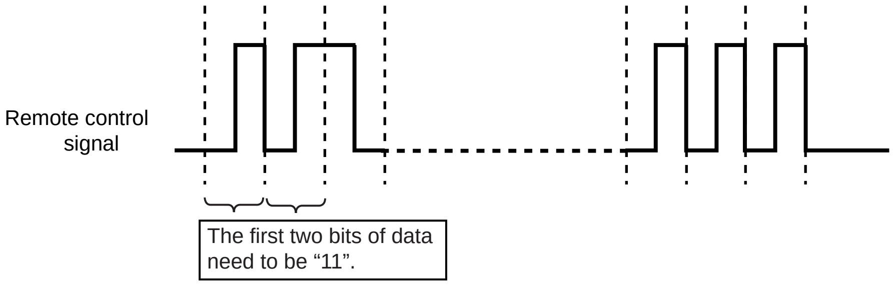


Figure 14-11 The waveform pattern in phase method

15. Analog/Digital Converter (ADC)

The TMPM3U6FY/FW contains a 12/10 (selectable) bit successive-approximation analog-to-digital convert-er (ADC).

External analog input pins (AIN0 to AIN17) can also be used as input/output ports.

15.1 Functions and features

1. It can select analog input and start AD conversion when receiving trigger signal from PMD(MPT) 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.

15.2 Block Diagram

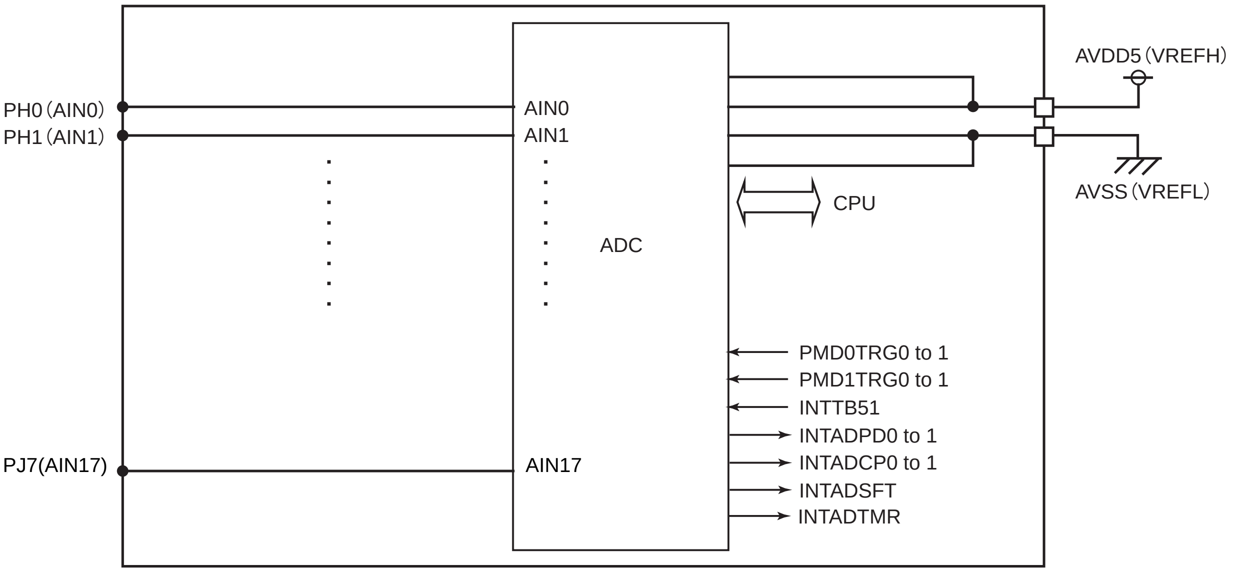


Figure 15-1 AD converters Block Diagram

15.3 List of Registers

Base Address = 0x4003_0000

Register Name		Address(Base+)
Clock Setting Register	ADCLK	0x0000
Mode Setting Register 0	ADMOD0	0x0004
Mode Setting Register 1	ADMOD1	0x0008
Mode Setting Register 2	ADMOD2	0x000C
Monitoring Setting Register 0	ADCMPCR0	0x0010
Monitoring Setting Register 1	ADCMPCR1	0x0014
Conversion Result Compare Register 0	ADCMP0	0x0018
Conversion Result Compare Register 1	ADCMP1	0x001C
Conversion Result Register 0	ADREG0	0x0020
Conversion Result Register 1	ADREG1	0x0024
Conversion Result Register 2	ADREG2	0x0028
Conversion Result Register 3	ADREG3	0x002C
Conversion Result Register 4	ADREG4	0x0030
Conversion Result Register 5	ADREG5	0x0034
Conversion Result Register 6	ADREG6	0x0038
Conversion Result Register 7	ADREG7	0x003C
Conversion Result Register 8	ADREG8	0x0040
Conversion Result Register 9	ADREG9	0x0044
Conversion Result Register 10	ADREG10	0x0048
Conversion Result Register 11	ADREG11	0x004C
PMD Trigger Program Number Select Register 0	ADPSEL0	0x0050
PMD Trigger Program Number Select Register 1	ADPSEL1	0x0054
PMD Trigger Program Number Select Register 2	ADPSEL2	0x0058
PMD Trigger Program Number Select Register 3	ADPSEL3	0x005C
Reserved	-	0x0060
Reserved	-	0x0064
Reserved	-	0x0068
Reserved	-	0x006C
Reserved	-	0x0070
Reserved	-	0x0074
Reserved	-	0x0078
Reserved	-	0x007C
PMD Trigger Interrupt Select Register 0	ADPINTS0	0x0080
PMD Trigger Interrupt Select Register 1	ADPINTS1	0x0084
PMD Trigger Interrupt Select Register 2	ADPINTS2	0x0088
PMD Trigger Interrupt Select Register 3	ADPINTS3	0x008C
PMD Trigger Interrupt Select Register 4	ADPINTS4	0x0090
PMD Trigger Interrupt Select Register 5	ADPINTS5	0x0094
PMD Trigger Program Register 0	ADPSET0	0x0098
PMD Trigger Program Register 1	ADPSET1	0x009C
PMD Trigger Program Register 2	ADPSET2	0x00A0
PMD Trigger Program Register 3	ADPSET3	0x00A4
PMD Trigger Program Register 4	ADPSET4	0x00A8
PMD Trigger Program Register 5	ADPSET5	0x00AC
Timer Trigger Program Registers 0 to 3	ADTSET03	0x00B0
Timer Trigger Program Registers 4 to 7	ADTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADTSET811	0x00B8

Base Address = 0x4003_0000

Register Name		Address(Base+)
Software Trigger Program Registers 0 to 3	ADSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADSSET811	0x00C4
Constant Conversion Program Registers0 to 3	ADASET03	0x00C8
Constant Conversion Program Registers 4 to 7	ADASET47	0x00CC
Constant Conversion Program Registers 8 to 11	ADASET811	0x00D0
Mode Setting Register 3	ADMOD3	0x00D4

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

15.4 Register Descriptions

AD conversion is performed at the clock frequency selected in the ADC Clock Setting Register.

15.4.1 ADCLK (Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-3	TSH[3:0]	R/W	Write as "1001".
2-0	ADCLK[2:0]	R/W	AD prescaler output (SCLK) select 000: fc (Note) 001 to 111: Reserved

Note: The AD conversion times are 1T = 74×(1/SCLK) in the 12-bit mode and T = 68×(1/SCLK) in the 10-bit mode.

15.4.2 ADMOD0 (Mode Setting Register 0)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	DACON	R/W	ADC operation control 1 0: Stop 1: Operate Setting <DACON> to "1", when using the ADC.
0	ADSS	W	Software triggered conversion 0: Don't care 1: Start Setting <ADSS> to "1" starts AD conversion (software triggered conversion). Receiving trigger signal from PMD (MPT) or TMRB(interrupt) starts AD conversion also. For detail setting, please read the chapter about PMD(MPT) and TMRB.

15.4.3 ADMOD1 (Mode Setting Register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	ADEN	R/W	AD conversion control 0: Disable 1: Enable Setting <ADEN> to "1", when using the ADC. After Setting <ADEN> to "1", setting <ADAS> to "1" starts AD conversion and repeat conversion.
6-1	−	R	Read as "0".
0	ADAS	R/W	Constant AD conversion control 0: Disable 1: Enable

15.4.4 ADMOD2 (Mode Setting Register 2)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	ADSFN	R	Software conversion busy flag 0: Conversion completed 1: Conversion in progress The <ADSFN> is a software AD conversion busy flag. After <ADSS> was set to "1", when AD conversion is actually started, <ADSFN> is set to "1". When finished AD conversion, <ADSFN> is cleared to "0".
0	ADBFN	R	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress The <ADBFN> is an AD conversion busy flag. When AD conversion is started regardless of conversion factor (PMD(MPT), Timer, Software, Constant), <ADBFN> is set to "1". When finished AD conversion, <ADBFN> is cleared to "0".

15.4.5 ADMOD3 (Mode Setting Register 3)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-12	−	R/W	Write as "0".
11-10	BITS[1:0]	R/W	12-bit/10-bit resolution mode selection 00: 10-bit 01: 12-bit 10 to 11: Reserved
9	−	R/W	Write as "0".
8	RCUT	R/W	ADC operation control 2 0: Operate 1: Stop Write "0" under AD conversion. By setting ADMOD3<RCUT> to "1", consumption current will be reduced.
7	−	R/W	Write as "0".
6	−	R/W	Write as "1".
5-3	PMODE[2:0]	R/W	Write as "100".
2-0	−	R/W	Write as "0".

Note: ADMOD3<PMODE[2:0]> must be set to "100". And do not change other bits in ADMOD3 register.

15.4.6 ADCMPCR0(Monitoring Setting Register 0)

After fixing the conversion result, the interrupt signal (INTADCP0) is generated.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT0[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons · · 15: After 16 comparisons The ADCMPCR0 and ADCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
7	CMP0EN	R/W	Monitoring function 0:Disable 1:Enable By setting <CMP0EN>="0"(disable), accumulated number of decision counts is cleared.
6-5	-	R	Read as "0".
4	ADBIG0	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register Compares whether a result of analog input is larger or smaller than the compare register. Every time AD conversion, which is set in the <REGS0[3:0]>, is complete, large/small decision is performed. If the result is matched the setting in <ADBIG0>, the counter increments.
3-0	REGS0[3:0]	R/W	AD conversion result register to be compared 0000: ADREG0 / 0100: ADREG4 / 1000: ADREG8 0001: ADREG1 / 0101: ADREG5 / 1001: ADREG9 0010: ADREG2 / 0110: ADREG6 / 1010: ADREG10 0011: ADREG3 / 0111: ADREG7 / 1011: ADREG11

15.4.7 ADCMPCR1(Monitoring Setting Register 1)

After fixing the conversion result, the interrupt signal (INTADCP0) is generated.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT1[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons · · 15: After 16 comparisons The ADCMPCR0 and ADCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
7	CMP1EN	R/W	Monitoring function 0:Disable 1:Enable By setting <CMP1EN>="0"(disable), accumulated number of decision counts is cleared.
6-5	-	R	Read as "0".
4	ADBIG1	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register Compares whether a result of analog input is larger or smaller than the compare register. Every time AD conversion, which is set in the <REGS1[3:0]>, is complete, large/small decision is performed. If the result is matched the setting in <ADBIG1>, the counter increments.
3-0	REGS1[3:0]	R/W	AD conversion result register to be compared 0000: ADREG0 0100: ADREG4 1000: ADREG8 0001: ADREG1 0101: ADREG5 1001: ADREG9 0010: ADREG2 0110: ADREG6 1010: ADREG10 0011: ADREG3 0111: ADREG7 1011: ADREG11

15.4.8 ADCMP0(Conversion Result Compare Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	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".

Note: Set the AD monitoring function to be prohibited (<CMP0EN>="0" , <CMP1EN> = "0") in advance-when this register is modified.

15.4.9 ADCMP1(Conversion Result Compare Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	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".

Note: Set the AD monitoring function to be prohibited (<CMP0EN>="0" , <CMP1EN> = "0") in advance-when this register is modified.

15.4.10 ADREG0(Conversion Result Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR0				-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR0[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR0	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG0 is read and is cleared when the ADREG0 is read.
0	ADR0RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR0RF> is a flag that is set when an AD conversion result is stored in the ADREG0 register and is cleared when the low-order byte of ADREG0 is read.

15.4.11 ADREG1(Conversion Result Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR1				-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR1[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR1	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG1 is read and is cleared when the ADREG1 is read.
0	ADR1RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR1RF> is a flag that is set when an AD conversion result is stored in the ADREG1 register and is cleared when the ADREG1 is read.

15.4.12 ADREG2(Conversion Result Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR2				-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR2[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR2	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG2 is read and is cleared when the ADREG2 is read.
0	ADR2RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR2RF> is a flag that is set when an AD conversion result is stored in the ADREG2 register and is cleared when the ADREG2 is read.

15.4.13 ADREG3(Conversion Result Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR3				-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR3[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR3	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG3 is read and is cleared when the ADREG3 is read.
0	ADR3RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR3RF> is a flag that is set when an AD conversion result is stored in the ADREG3 register and is cleared when the ADREG3 is read.

15.4.14 ADREG4(Conversion Result Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR4				-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR4[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR4	R	OverrunOverrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG4 is read and is cleared when the ADREG4 is read.
0	ADR4RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR4RF> is a flag that is set when an AD conversion result is stored in the ADREG4 register and is cleared when the ADREG4 is read.

15.4.15 ADREG5(Conversion Result Register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR5				-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR5[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR5	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG5 is read and is cleared when the ADREG5 is read.
0	ADR5RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR5RF> is a flag that is set when an AD conversion result is stored in the ADREG5 register and is cleared when the ADREG5 is read.

15.4.16 ADREG6(Conversion Result Register 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR6							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR6				-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR6[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR6	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG6 is read and is cleared when the ADREG6 is read.
0	ADR6RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR6RF> is a flag that is set when an AD conversion result is stored in the ADREG6 register and is cleared when the ADREG6 is read.

15.4.17 ADREG7(Conversion Result Register 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR7							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR7				-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR7[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR7	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG7 is read and is cleared when the ADREG7 is read.
0	ADR7RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR7RF> is a flag that is set when an AD conversion result is stored in the ADREG7 register and is cleared when the ADREG7 is read.

15.4.18 ADREG8(Conversion Result Register 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR8							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR8				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR8[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR8	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG8 is read and is cleared when the ADREG8 is read.
0	ADR8RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR8RF> is a flag that is set when an AD conversion result is stored in the ADREG8 register and is cleared when the ADREG8 is read.

15.4.19 ADREG9(Conversion Result Register 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR9							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR9				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR9[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR9	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG9 is read and is cleared when the ADREG9 is read.
0	ADR9RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR9RF> is a flag that is set when an AD conversion result is stored in the ADREG9 register and is cleared when the ADREG9 is read.

15.4.20 ADREG10(Conversion Result Register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR10[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR10	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG10 is read and is cleared when the ADREG10 is read.
0	ADR10RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR10RF> is a flag that is set when an AD conversion result is stored in the ADREG10 register and is cleared when the ADREG10 is read.

15.4.21 ADREG11(Conversion Result Register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR11							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR11				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR11[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR11	R	Overrun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADREG11 is read and is cleared when the ADREG11 is read.
0	ADR11RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR11RF> is a flag that is set when an AD conversion result is stored in the ADREG11 register and is cleared when the ADREG11 is read.

15.4.22 PMD Trigger Program Registers

The AD converter can be started by a trigger signal generated in the PMD (programmable motor driver) circuit (MPT circuit in the PMD mode).

The PMD trigger program registers are used to specify the program to be started by each of four 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.

- PMD Trigger Program Number Select Register (ADPSEL0 to ADPSEL3)

  The PMD Trigger Program Number Select Register (ADPSELn) specifies the program to be started by each of four AD conversion start signals corresponding to twelve triggers(PMD0TRG0 to 1 , PMD1TRG0 to 1) generated by the PMD Programs 0 to 5 are available.

  "ADxPSEL0 to ADxPSEL1" corresponds to "PMD0TRG0 to 1". "ADPSEL2 to ADPSEL3" corresponds to "PMD1TRG0 to 1".
- PMD Trigger Interrupt Select Register (ADPINTS0 to ADPINTS5)

  The PMD Trigger Interrupt Select Registers (ADPINTS0 to ADPINTS5) select the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

  ADPINTS0 corresponds to program 0, and it exists to ADPINT5 (program 5).
- PMD Trigger Program Register (ADPSET0 to ADPSET5)

  The PMD Trigger Program Setting Registers (ADPSET0 to ADPSET5) 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 ADPSETn0 to ADPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADREG0 to ADREG3).

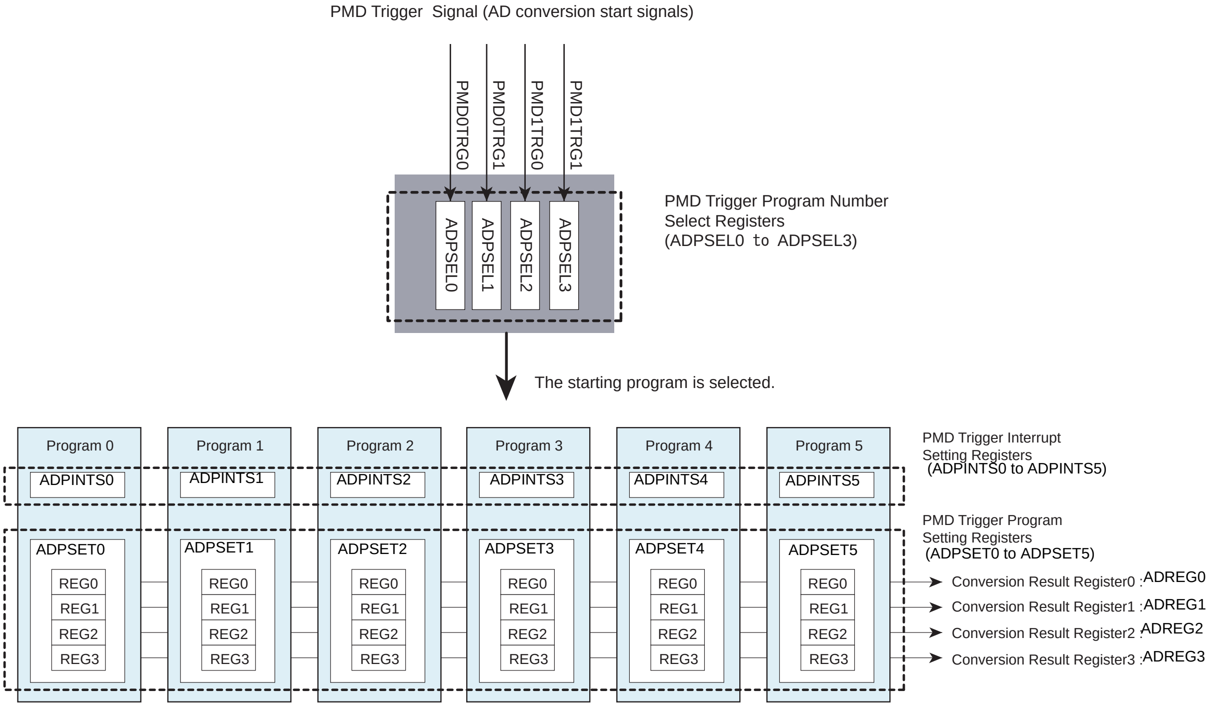

PMD Trigger Signal (AD conversion start signals)
PMD0TRG0
PMD0TRG1
PMD1TRG0
PMD1TRG1
ADPSEL0
ADPSEL1
ADPSEL2
ADPSEL3
PMD Trigger Program Number Select Registers (ADPSEL0 to ADPSEL3)
The starting program is selected.
Program 0
Program 1
Program 2
Program 3
Program 4
Program 5
ADPINTS0
ADPINTS1
ADPINTS2
ADPINTS3
ADPINTS4
ADPINTS5
PMD Trigger Interrupt Setting Registers (ADPINTS0 to ADPINTS5)
ADPSET0
ADPSET1
ADPSET2
ADPSET3
ADPSET4
ADPSET5
PMD Trigger Program Setting Registers (ADPSET0 to ADPSET5)
REG0
REG1
REG2
REG3
Conversion Result Register0 :ADREG0
Conversion Result Register1 :ADREG1
Conversion Result Register2 :ADREG2
Conversion Result Register3 :ADREG3

Figure 15-2 PMD Trigger Program Registers

15.4.22.1 ADPSEL0 to ADPSEL3(PMD Trigger Program Number Select Register 0 to 3)

ADPSEL0:PMD Trigger Program Number 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	PENS0	-	-	-	-	PMDS0		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS0	R/W	PMD0TRG0 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS0[2:0]	R/W	Program number select (Refer to Table 15-1)

ADPSEL1:PMD Trigger Program Number 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	PENS1	-	-	-	-	PMDS1		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS1	R/W	PMD0TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS1[2:0]	R/W	Program number select (Refer to Table 15-1)

ADPSEL2:PMD Trigger Program Number 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	PENS2	-	-	-	-	PMDS2		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS2	R/W	PMD1TRG0 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS2[2:0]	R/W	Program number select (Refer to Table 15-1)

ADPSEL3:PMD Trigger Program Number 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	PENS3	-	-	-	-	PMDS3		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS3	R/W	PMD1TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS3[2:0]	R/W	Program number select (Refer to Table 15-1)

Table 15-1 Program number select

<PMDS0[2:0]> to <PMDS3[2:0]>	
000	Program0
001	Program1
010	Program2
011	Program3
100	Program4
101	Program5
110	reserved
111	reserved

15.4.22.2 ADPINTS0 to 5(PMD Trigger Interrupt Select Register 0 to 5)

ADPINTS0: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:INTADPD0 10:INTADPD1 11: No interrupt output The starting interrupt is selected for program 0.

ADPINTS1: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:INTADPD0 10:INTADPD1 11: No interrupt output The starting interrupt is selected for program 1.

ADPINTS2: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:INTADPD0 10:INTADPD1 11: No interrupt output The starting interrupt is selected for program 2.

ADPINTS3: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:INTADPD0 10:INTADPD1 11: No interrupt output The starting interrupt is selected for program 3.

ADPINTS4: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:INTADPD0 10:INTADPD1 11: No interrupt output The starting interrupt is selected for program 4.

ADPINTS5: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:INTADPD0 10:INTADPD1 11: No interrupt output The starting interrupt is selected for program 5.

15.4.22.3 ADPSET0 to 5(PMD Trigger Program Register 0 to 5)

Each ADPSETn (n=0 to 5:Program number) is composed of four sets that assume <AINSPnm [4:0]>, <UVWISnm[1:0]>, and <ENSPnm> in a couple.

(m=0 to 3)

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 15-2 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	AD Channel
0_0000	:AIN0
0_0001	:AIN1
0_0010	:AIN2
0_0011	:AIN3
0_0100	:AIN4
0_0101	:AIN5
0_0110	:AIN6
0_0111	:AIN7
0_1000	:AIN8
0_1001	:AIN9
0_1010	:AIN10
0_1011	:AIN11
0_1100	:AIN12
0_1101	:AIN13
0_1110	:AIN14
0_1111	:AIN15
1_0000	:AIN16
1_0001	:AIN17
1_0010 to 1_1111	:reserved

ADPSET0: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	ADREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write as "0".
28-24	AINSP03[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
23	ENSP02	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write as "0".
20-16	AINSP02[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
15	ENSP01	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write as "0".
12-8	AINSP01[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
7	ENSP00	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write as "0".
4-0	AINSP00[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".

ADPSET1: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	ADREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write as "0".
28-24	AINSP13[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
23	ENSP12	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write as "0".
20-16	AINSP12[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
15	ENSP11	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write as "0".
12-8	AINSP11[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
7	ENSP10	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write as "0".
4-0	AINSP10[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".

ADPSET2: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	ADREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write as "0".
28-24	AINSP23[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
23	ENSP22	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write as "0".
20-16	AINSP22[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
15	ENSP21	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write as "0".
12-8	AINSP21[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
7	ENSP20	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write as "0".
4-0	AINSP20[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".

ADPSET3: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	ADREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write as "0".
28-24	AINSP33[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
23	ENSP32	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write as "0".
20-16	AINSP32[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
15	ENSP31	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write as "0".
12-8	AINSP31[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
7	ENSP30	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write as "0".
4-0	AINSP30[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".

ADPSET4: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	ADREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write as "0".
28-24	AINSP43[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
23	ENSP42	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write as "0".
20-16	AINSP42[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
15	ENSP41	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write as "0".
12-8	AINSP41[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
7	ENSP40	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write as "0".
4-0	AINSP40[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".

ADPSET5: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	ADREG3 enable 0:Disable 1:Enable
30-29	-	R/W	Write as "0".
28-24	AINSP53[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
23	ENSP52	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R/W	Write as "0".
20-16	AINSP52[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
15	ENSP51	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R/W	Write as "0".
12-8	AINSP51[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".
7	ENSP50	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R/W	Write as "0".
4-0	AINSP50[4:0]	R/W	AIN select Refer to "Table 15-2 Select the AIN pin".

15.4.23 ADTSET03 / ADTSET47 / ADTSET811 (Timer Trigger Program Registers)

AD conversion can be started by INTTB51 generated from Timer5(TMRB5) as a trigger. There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to "1" enables the ADTSETm register. The <AINSTm[4:0]> are used to select the AIN pin to be used. The numbers of the Timer Trigger Program Registers correspond to those of the AD Conversion Result Registers. When finished this AD conversion, interrupt : INTADTMR is generated.

(m=0 to 11)

Table 15-3 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	AD Channel
0_0000	:AIN0
0_0001	:AIN1
0_0010	:AIN2
0_0011	:AIN3
0_0100	:AIN4
0_0101	:AIN5
0_0110	:AIN6
0_0111	:AIN7
0_1000	:AIN8
0_1001	:AIN9
0_1010	:AIN10
0_1011	:AIN11
0_1100	:AIN12
0_1101	:AIN13
0_1110	:AIN14
0_1111	:AIN15
1_0000	:AIN16
1_0001	:AIN17
1_0010 to 1_1111	:reserved

ADTSET03: Timer Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST3	R/W	ADREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST3[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
23	ENST2	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST2[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
15	ENST1	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST1[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
7	ENST0	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST0[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".

ADTSET47: Timer Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST6	-	-	AINST6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST5	-	-	AINST5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST4	-	-	AINST4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST7	R/W	ADREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST7[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
23	ENST6	R/W	ADREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST6[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
15	ENST5	R/W	ADREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST5[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
7	ENST4	R/W	ADREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST4[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".

ADTSET811: Timer Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST10	-	-	AINST10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST9	-	-	AINST9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST8	-	-	AINST8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST11	R/W	ADREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST11[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
23	ENST10	R/W	ADREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST10[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
15	ENST9	R/W	ADREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST9[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".
7	ENST8	R/W	ADREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST8[4:0]	R/W	AIN select Refer to "Table 15-3 Select the AIN pin".

15.4.24 ADSSET03 / ADSSET47 / ADSSET811(Software Trigger Program Registers)

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers. Setting the <ENSSm> to "1" enables the ADSSETm register. The <AINSSm[4:0]> are used to select the AIN pin to be used. When finished this AD conversion, interrupt :INTADSFT is generated.

(m=0 to 11)

Table 15-4 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	AD Channel
0_0000	:AIN0
0_0001	:AIN1
0_0010	:AIN2
0_0011	:AIN3
0_0100	:AIN4
0_0101	:AIN5
0_0110	:AIN6
0_0111	:AIN7
0_1000	:AIN8
0_1001	:AIN9
0_1010	:AIN10
0_1011	:AIN11
0_1100	:AIN12
0_1101	:AIN13
0_1110	:AIN14
0_1111	:AIN15
1_0000	:AIN16
1_0001	:AIN17
1_0010 to 1_1111	:reserved

ADSSET03: Software Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS2	-	-	AINSS2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS1	-	-	AINSS1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS0	-	-	AINSS0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS3	R/W	ADREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS3[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
23	ENSS2	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS2[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
15	ENSS1	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS1[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
7	ENSS0	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS0[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".

ADSSET47: Software Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS6	-	-	AINSS6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS5	-	-	AINSS5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS4	-	-	AINSS4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS7	R/W	ADREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS7[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
23	ENSS6	R/W	ADREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS6[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
15	ENSS5	R/W	ADREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS5[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
7	ENSS4	R/W	ADREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS4[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".

ADSSET811: Software Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS10	-	-	AINSS10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS9	-	-	AINSS9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS8	-	-	AINSS8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS11	R/W	ADREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS11[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
23	ENSS10	R/W	ADREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS10[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
15	ENSS9	R/W	ADREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS9[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".
7	ENSS8	R/W	ADREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS8[4:0]	R/W	AIN select Refer to "Table 15-4 Select the AIN pin".

15.4.25 ADASET03 / ADASET47 / ADASET811(Constant Conversion Program Registers)

The ADCs allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to "1" enables the ADASETm register. The <AINSAm[4:0]> are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m=0 to 11)

Table 15-5 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	AD Channel
0_0000	:AIN0
0_0001	:AIN1
0_0010	:AIN2
0_0011	:AIN3
0_0100	:AIN4
0_0101	:AIN5
0_0110	:AIN6
0_0111	:AIN7
0_1000	:AIN8
0_1001	:AIN9
0_1010	:AIN10
0_1011	:AIN11
0_1100	:AIN12
0_1101	:AIN13
0_1110	:AIN14
0_1111	:AIN15
1_0000	:AIN16
1_0001	:AIN17
1_0010 to 1_1111	:reserved

ADASET03: Constant Conversion Program Registers03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA2	-	-	AINSA2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA1	-	-	AINSA1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA0	-	-	AINSA0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA3	R/W	ADREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA3[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
23	ENSA2	R/W	ADREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA2[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
15	ENSA1	R/W	ADREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA1[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
7	ENSA0	R/W	ADREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA0[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".

ADASET47: Constant Conversion Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA6	-	-	AINSA6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA5	-	-	AINSA5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA4	-	-	AINSA4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA7	R/W	ADREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA7[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
23	ENSA6	R/W	ADREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA6[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
15	ENSA5	R/W	ADREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA5[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
7	ENSA4	R/W	ADREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA4[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".

ADASET811: 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	ADREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA11[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
23	ENSA10	R/W	ADREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA10[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
15	ENSA9	R/W	ADREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA9[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".
7	ENSA8	R/W	ADREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA8[4:0]	R/W	AIN select Refer to "Table 15-5 Select the AIN pin".

15.5 Operation Descriptions

15.5.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the AVDD5and AVSSpins are used . If ADMOD3<RCUT> is set to "1", current flowing between AVDD5 and AVSS can be controlled to reduce consumption current. When AD converter is used, ADMOD3<RCUT> is set to "0".

Note 1: During AD conversion, do not change the output data of port H/I/J, to avoid the influence on the conversion result.

Note 2: AD conversion results might be unstable by the following conditions.
Input operation is executed.
Output operation is executed.
Output current of port varies.
Take a countermeasure such as averaging the multiple conversion results, to get precise value.

15.5.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD(MPT) trigger
- Timer trigger
- Software trigger

These start triggers are given priorities as shown below.

PMD trigger 0 > ···· > PMD trigger 3 > Timer trigger > Software trigger > constant trigger

When a higher-priority trigger occurs while an AD conversion is in progress, a higher-priority trigger is handled stop the ongoing program and start AD conversion correspond to a higher-priority trigger number.

When the PMD trigger occurs while a 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.

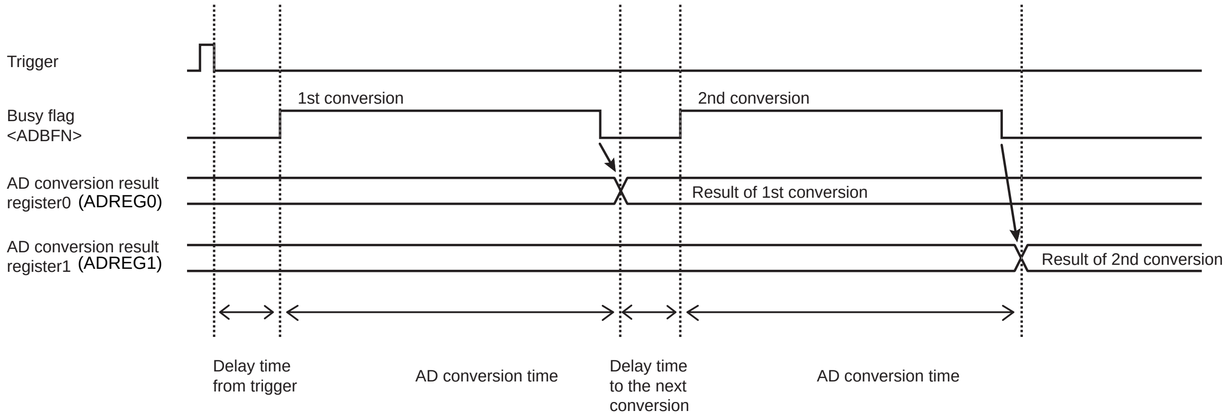


Figure 15-3 Timing chart of AD conversion

Table 15-6 AD conversion time and delay time (SCLK = 40MHz, unit : μs)

		fsys = 40MHz	
	Trigger	MIN	MAX
Delay time from trigger (Note 1)	PMD (MPT)	0.225	0.3
	TMRB	0.225	0.5
	Software, Constant	0.25	0.525
AD conversion time	−	1.85	
Delay time to the next conversion (Note2)	PMD (MPT)	0.175	0.225
	TMRB, Software, Constant	0.175	0.425

Note 1: Delay time from trigger to start of AD conversion.

Note 2: Delay time to the 2nd or after conversion in plural conversions with one trigger.

15.5.3 AD Conversion Monitoring Function

The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function, set ADCMPCR0<CMP0EN> or ADCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by ADCMPCR0<ADBIG0>, the interrupt (INTADCPA for ADCMPCR0, INTADCPB for ADCMPCR1) is generated. The comparison is executed at the timing of storing the conversion result into the register.

Note 1: The AD conversion result store flag <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.

15.6 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

15.6.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADSSET03, ADSSET47 and ADSSET811.(Figure 15-4)

If the ADMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.(Figure 15-5)

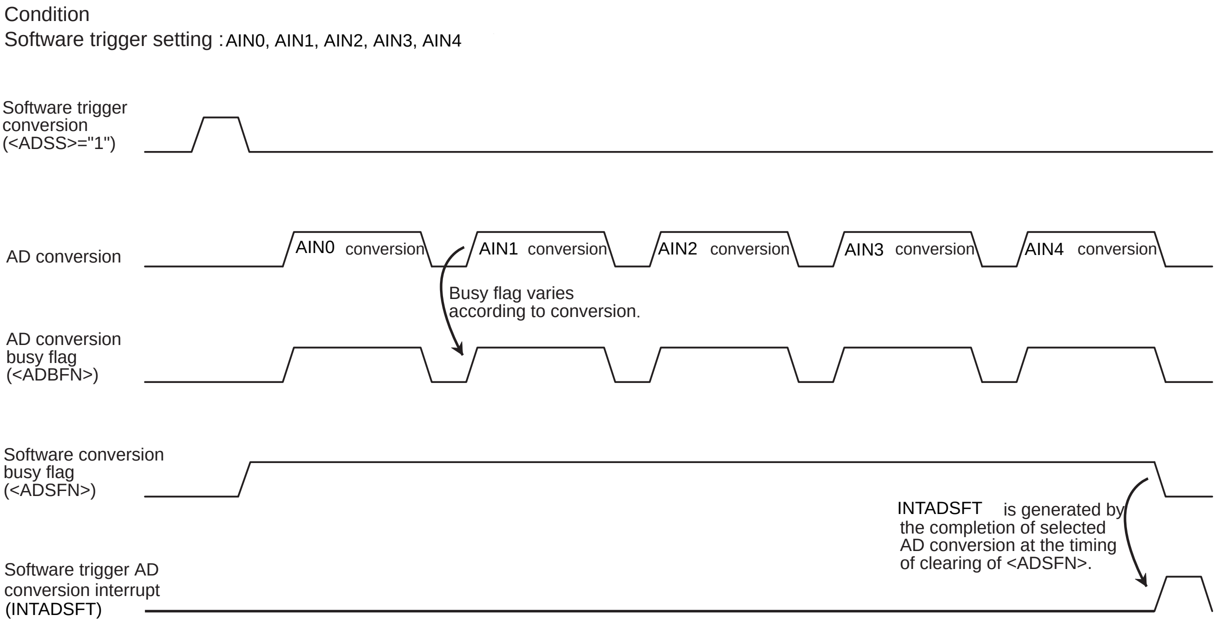

Condition
Software trigger setting : AIN0, AIN1, AIN2, AIN3, AIN4
Software trigger conversion (<ADSS>="1")
AD conversion
AIN0 conversion
AIN1 conversion
AIN2 conversion
AIN3 conversion
AIN4 conversion
Busy flag varies according to conversion.
AD conversion busy flag (<ADBFN>)
Software conversion busy flag (<ADSFN>)
INTADSFT is generated by the completion of selected AD conversion at the timing of clearing of <ADSFN>.
Software trigger AD conversion interrupt (INTADSFT)

Figure 15-4 Software trigger AD conversion

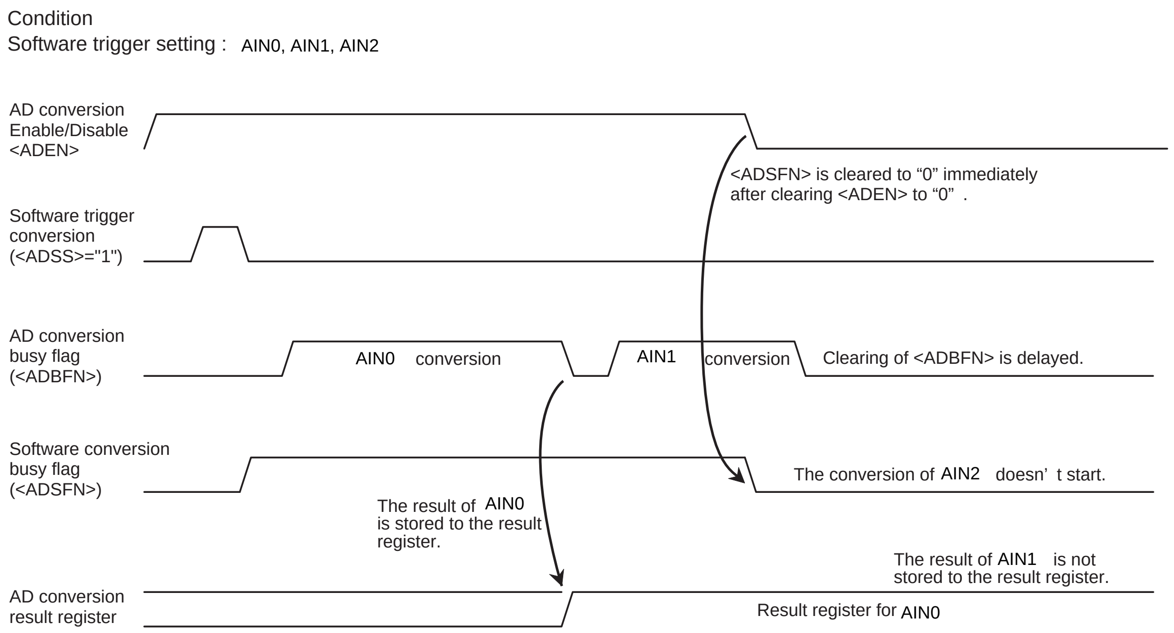

Condition
Software trigger setting : AIN0, AIN1, AIN2
AD conversion Enable/Disable <ADEN>
<ADSFN> is cleared to “0” immediately after clearing <ADEN> to “0” .
Software trigger conversion (<ADSS>="1")
AD conversion busy flag (<ADBFN>)
AIN0 conversion
AIN1 conversion
Clearing of <ADBFN> is delayed.
Software conversion busy flag (<ADSFN>)
The conversion of AIN2 doesn’ t start.
The result of AIN0 is stored to the result register.
The result of AIN1 is not stored to the result register.
AD conversion result register
Result register for AIN0

Figure 15-5 Writing "0" to <ADEN> during the software trigger AD conversion

15.6.2 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 15-6)

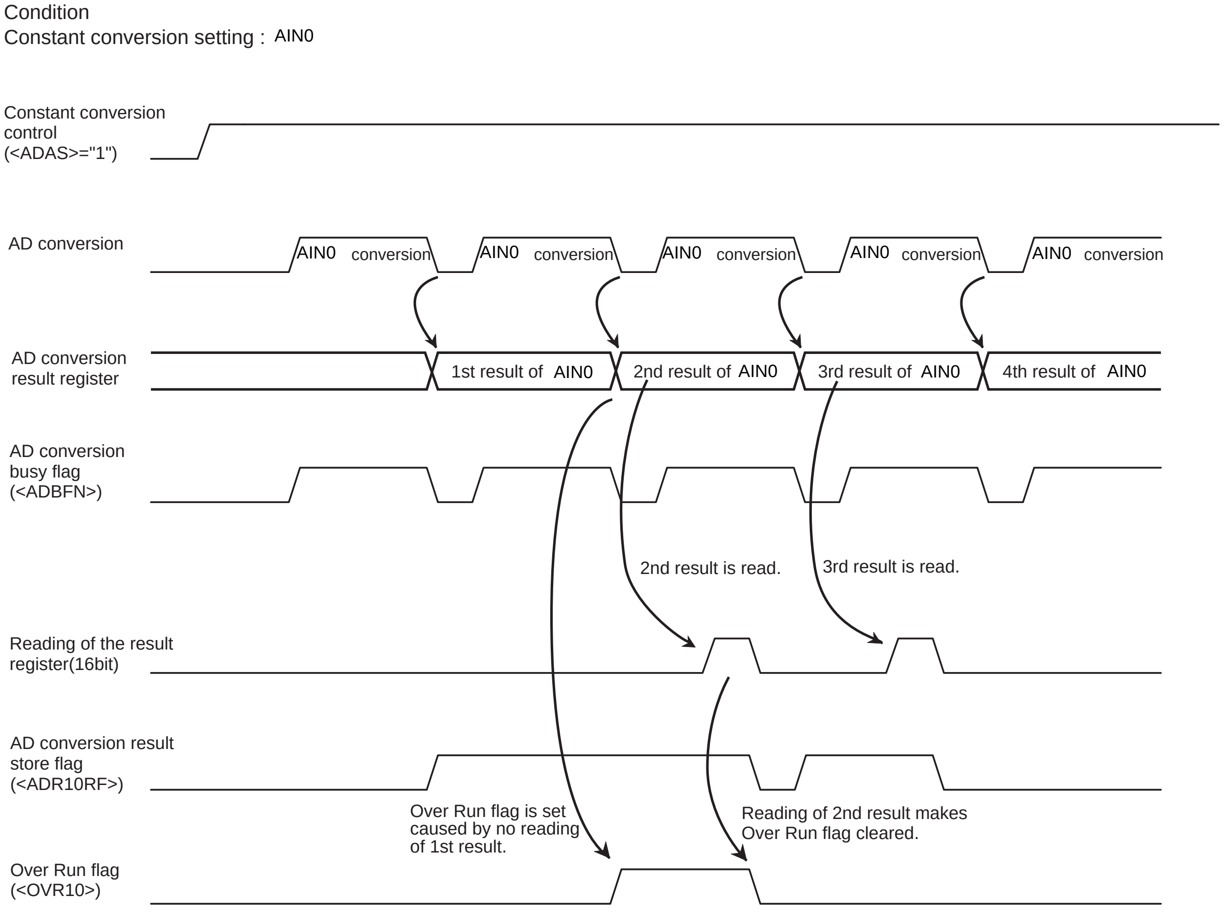

Condition
Constant conversion setting : AIN0
Constant conversion control (<ADAS>="1")
AD conversion
AIN0 conversion
AIN0 conversion
AIN0 conversion
AIN0 conversion
AIN0 conversion
AD conversion result register
1st result of AIN0
2nd result of AIN0
3rd result of AIN0
4th result of AIN0
AD conversion busy flag (<ADBFN>)
2nd result is read.
3rd result is read.
Reading of the result register(16bit)
AD conversion result store flag (<ADR10RF>)
Over Run flag is set caused by no reading of 1st result.
Reading of 2nd result makes Over Run flag cleared.
Over Run flag (<OVR10>)

Figure 15-6 Constant conversion

15.6.3 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 15-7) 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 15-8)

Condition
Software trigger setting : AIN0, AIN1, AIN2
PMD trigger setting : AIN0

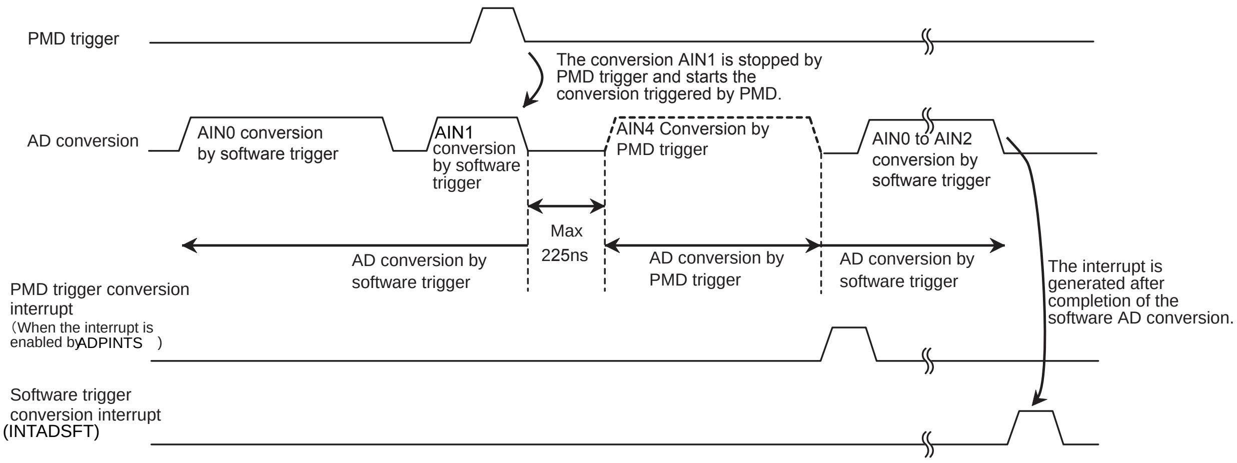

PMD trigger
The conversion AIN1 is stopped by PMD trigger and starts the conversion triggered by PMD.
AD conversion
AIN0 conversion by software trigger
AIN1 conversion by software trigger
AIN4 Conversion by PMD trigger
AIN0 to AIN2 conversion by software trigger
Max 225ns
AD conversion by software trigger
AD conversion by PMD trigger
AD conversion by software trigger
PMD trigger conversion interrupt
（When the interrupt is enabled byADPINTS　)
The interrupt is generated after completion of the software AD conversion.
Software trigger conversion interrupt (INTADSFT)

Figure 15-7 AD conversion by PMD trigger

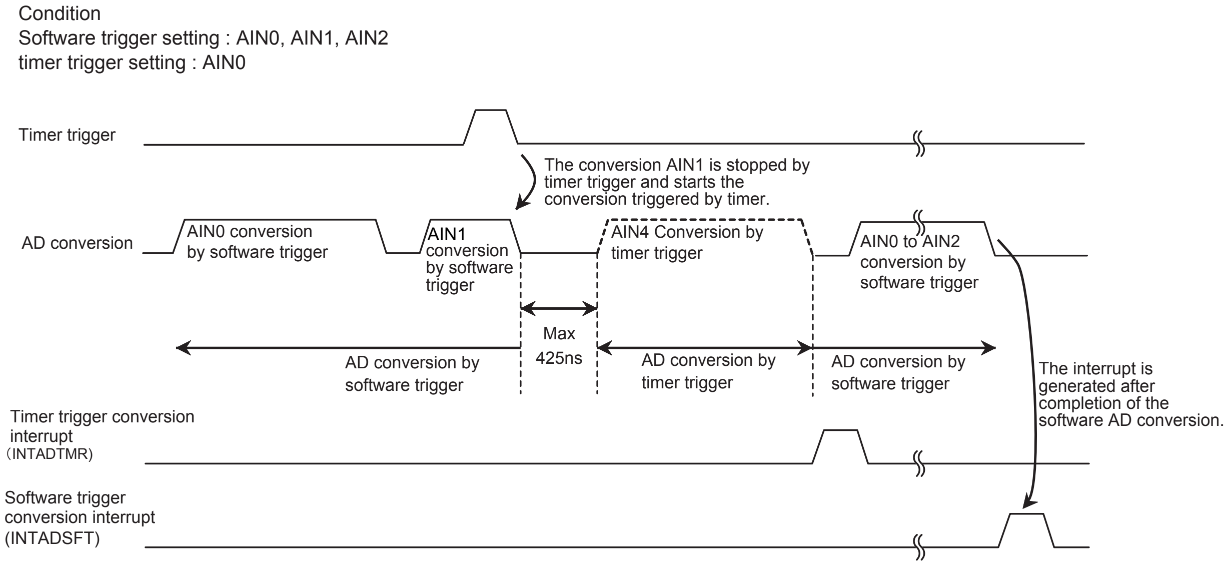

Condition
Software trigger setting : AIN0, AIN1, AIN2
timer trigger setting : AIN0
Timer trigger
The conversion AIN1 is stopped by timer trigger and starts the conversion triggered by timer.
AD conversion
AIN0 conversion by software trigger
AIN1 conversion by software trigger
AIN4 Conversion by timer trigger
AIN0 to AIN2 conversion by software trigger
Max 425ns
AD conversion by software trigger
AD conversion by timer trigger
AD conversion by software trigger
The interrupt is generated after completion of the software AD conversion.
Timer trigger conversion interrupt (INTADTMR)
Software trigger conversion interrupt (INTADSFT)

Figure 15-8 AD conversion by timer trigger

Note: When Timer trigger is not use, do not use INTTB51.

16. Encoder Input Circuit (ENC)

16.1 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

16.2 Differences between channels

The TMPM3U6FY/FW has a two-channel incremental encoder interface (ENC0 and ENC1), which can ob-tain the absolute position of the motor, based on input signals from the incremental encoder.

These channels operate identical except the differences in below.

Table 16-1 Differences between channels

Channel	Input pin			Encoder input in-terrupt
	A-phase	B-phase	Z-phase	
Channel0	PD0 / ENCA0	PD1 / ENCB0	PD2 / ENCZ0	INTENC0
Channel1	PF2 / ENCA1	PF3 / ENCB1	PF4 / ENCZ1	INTENC1

16.3 Block Diagram

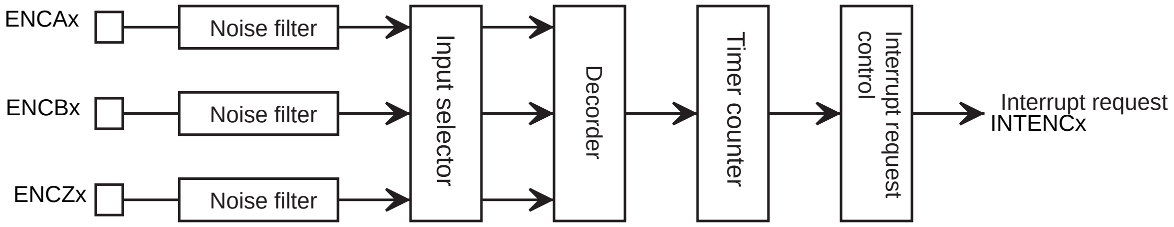


Figure 16-1 Block diagram of encoder input circuit

16.4 Registers

16.4.1 List of Registers

The following is control registers and addresses of encoder input circuit.

Channel x	Base Address
Channel0	0x4001_0400
Channel1	0x4001_0500

Register name (x=0,1)		Address (Base+)
Encoder Input Control Register	ENxTNCR	0x0000
Encoder Counter Reload Register	ENxRELOAD	0x0004
Encoder Compare Register	ENxINT	0x0008
Encoder Counter	ENxCNT	0x000C

16.4.2 ENxTNCR (Encoder Input Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	MODE		P3EN
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ZEN	ENRUN	NR		INTEN	ENDEV		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	−	R	Read as "0".
18-17	MODE[1:0]	R/W	Encoder input mode setting 00:Encoder mode 01:Sensor mode (event count)) 10:Sensor mode (timer count)) 11:Timer mode
16	P3EN	R/W	2-phase / 3-phase input selection (sensor mode) (Note 1) 0:2-phase input 1:3-phase input Sets the number of input signals.
15	CMP	R	Compare flag 0:- 1:Compare (Clear by RD) If comparing is executed, <CMP> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect.
14	REVERR	R	Reverse error flag (Sensor mode (at timer count)) (Note 2) 0:- 1:Error (Clear by RD) In sensor mode (at timer count), when a reverse error occurs, <REVERR> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect. In the encoder mode, sensor mode (event count) and timer mode, this bit has no meaning.
13	UD	R	Rotation direction 0:CCW (A-phase has the 90-degree phase lead to B-phase using incremental encoder) 1:CW (A-phase has the 90-degree phase lag to B-phase using incremental encoder) <UD> is set to "0", when <ENRUN> = 0.

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th colspan="2">Function</th></tr>
<tr><td>12</td><td>ZDET</td><td>R</td><td colspan="2">Z-Detected
0:Not detected
1:Z-phase detected

<ZDET> is set to 1 on the first edge of Z input signal (ENCZ) after <ENRUN> is written from 0 to 1.
This occurs on a rising edge of the signal Z during CW rotation or on a falling edge of Z during CCW rotation.
<ZDET> is set to "0" when <ENRUN> = 0.
<ZEN> has no influence on the value of <ZDET>.
<ZDET> is set to "0" in the sensor event count and the sensor timer count modes.
In the sensor mode (event count) and sensor mode (timer count), this bit is always set to "0".</td></tr>
<tr><td>11</td><td>SFTCAP</td><td>W</td><td colspan="2">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 ENxCNT register.
Writing "0" to <SFTCAP> has no effect. Reading <SFTCAP> always returns to "0".
In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; writing "1" to this bit is ignored.</td></tr>
<tr><td>10</td><td>ENCLR</td><td>W</td><td colspan="2">Encoder pulse counter clear
0:-
1:Clear

Writing a 1 to <ENCLR> clears the encoder counter to 0. Once cleared, the encoder counter restarts counting from 0. Writing "0" to <ENCLR> has no effect. Reading <ENCLR> always returns to "0".</td></tr>
<tr><td>9</td><td>ZESEL</td><td>R/W</td><td colspan="2">Edge selection of ENCZ (timer mode)
0:Rising edge
1:Falling edge

In timer mode, this bit selects inputs edge of ENCZ used as external trigger.
In the other mode, this bit has no meaning.</td></tr>
<tr><td>8</td><td>CMPEN</td><td>R/W</td><td colspan="2">Compare enable
0:Disable
1:Enable

When "1" is set to <CMPEN>, this bit compares counter values of encoder counter with register value of ENINT. When "0" is set to <CMPEN>, this compare is disabled.</td></tr>
<tr><td rowspan="3">7</td><td rowspan="3">ZEN</td><td rowspan="3">R/W</td><td colspan="2">Z-phase enable (Encoder mode/timer mode)
0:Disable
1:Enable

In the other mode, this bit has no meaning</td></tr>
<tr><td>&lt;Encoder mode>
Clear setting of encoder counter using ENCZ input</td><td>When &lt;ZEN> = "1" is set, if a rising edge of ENCZ is detected during rotating clockwise, the encoder counter is cleared to "0".
If a falling edge of ENCZ is detected during rotating counter- clockwise, the encoder counter is cleared to "0".
If the edges of ENCLK (multiply by 4 clock derived from the decoded A and B signals) and the edge of ENCZ coincide, the encoder counter is cleared to 0 without incrementing or decrementing (i.e., the clear takes precedence).</td></tr>
<tr><td>&lt;Timer mode>
Sets ENCZ input to use as an external trigger.</td><td>When &lt;ZEN> = 1, the value of the encoder counter is captured into the ENxINT register and cleared to 0 on the edge of ENCZ selected by <ZESEL&gt;.</td></tr>
<tr><td>6</td><td>ENRUN</td><td>R/W</td><td colspan="2">Encoder operation enable
0:Disable
1:Enable

Setting <ENRUN> to 1 and clearing <ZDET> to 0 enables the encoder operation.
Clearing <ENRUN> to 0 disables the encoder operation.
There are counters and flags that are cleared and not cleared when <ENRUN> bit is cleared to 0.</td></tr>
</table>

Bit	Bit Symbol	Type	Function
5-4	NR[1:0]	R/W	Noise filter 00:No filtering 01:Filters out pulses narrower than 31/fsys as noise (775ns@40MHz) 10:Filters out pulses narrower than 63/fsys as noise (1575ns@40MHz) 11:Filters out pulses narrower than 127/fsys as noise (3175ns@40MHz) The digital noise filters remove pulses narrower than the width selected by <NR[1:0]>.
3	INTEN	R/W	Encoder interrupt enable 0:Disable 1:Enable <INTEN> enables or disables the ENC interrupt. Setting <INTEN> to "1" enables interrupt generation. Setting <INTEN> to "0" disables interrupt generation.
2-0	ENDEV[2:0]	R/W	Encoder pulse division factor 000:divided by 1 100:divided by 16 001:divided by 2 101:divided by 32 010:divided by 4 110:divided by 64 011:divided by 8 111:divided by 128 Sets encoder pulse division factor The frequency of the encoder pulse is divided by the factor specified by <ENDEV[2:0]>. The divided signal determines the interval of the event interrupt.

Note 1: In the encoder mode or timer mode, <P3EN> must be set to "0".

Note 2: If changing the mode, first read the flag to clear.

The operation mode has eight modes specified with<MODE[1:0]>, <P3EN> and <ZEN>.

The operation mode settings are as follows:

<MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Mode
00	0	0	A, B	Encoder mode
	1		A,B,Z	Encoder mode (use of Z)
01	0	0	U,V	Sensor mode (event count, 2-phase input)
		1	U,V,W	Sensor mode (event count, 3-phase input)
10	0	0	U,V	Sensor mode (timer count, 2-phase input)
		1	U,V,W	Sensor mode (timer count, 3-phase input)
11	0	0	-	Timer mode
	1		Z	Timer mode (use of Z)

The following is the status of <ENRUN> and corresponding signals.

Counter/flag	<ENRUN> = 0 (After reset)	<ENRUN> = 1 (Operating)	<ENRUN> = 0 (Stopping)	<ENRUN> = 0 Object flag/counter clear procedure
Encoder counter	0x000000	Count operation	Maintains a value when stopping	Software clear (<ENCLR> = 1 WR)
Noise filter counter	000_0000	Count-up operation	Count-up operation (Always filtering)	Only reset
Encoder pulse division counter	0x00	Count-down operation	Stopped and cleared	Clear when <ENRUN> = 0
Compare flag <CMP>	0	"1" is set when comparing Clear when read.	Cleared	Clear when <ENRUN> = 0
Reverse error flag <REVERR>	0	"1" is set when error occurs. Clear when read.	Cleared	Clear when <ENRUN> = 0
Z detection flag <ZDET>	0	"1" is set when Z is detected.	Cleared	Clear when <ENRUN> = 0
Rotation direction bit <UD>	0	"0"/"1" is set depending on the direction	Cleared	Clear when <ENRUN> = 0

16.4.3 ENxRELOAD (Encoder Counter Reload Register)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	RELOAD[15:0]	R/W	Sets the Encoder counter period (after multiplied by 4 or 6) 0x0000 to 0xFFFF Z-phase is used : Sets the number of count pulses for one rotation Z-phase is not used : Sets the number of count pulses minus one for one rotation <RELOAD[15:0]> defines the encoder counter period multiplied by 4. If the encoder counter is configured as an up-counter, it increments up to the value programmed in <RELOAD[15:0]> and then wraps around to 0 on the next ENCLK. If the encoder counter is configured as a down-counter, it decrements to 0 and then is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.

The RELOAD register is only used in Encoder mode.

16.4.4 ENxINT (Encoder Compare Register)

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

Bit	Bit Symbol	Type	Function		
31-24	−	R	Read as "0".		
23-0	INT[23:0]	R/W	Counter compare value setting		
			Encoder mode:	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = 1 is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = 1is set, an interrupt request (INTENCx) occurs. However if <ZEN> = 1 is set, an interrupt request does not occur until <ZDET> = 1.	
			Sensor mode: (event count)	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = 1 is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = 1is set, an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.	
			Sensor mode: (Timer count)	Interrupt condition of abnormal pulse detection time	0x000000 to 0xFFFFFF
				When <CMPEN> = 1 is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = 1 is set, an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.	
			Timer mode	Interrupt condition of timer compare	0x000000 to 0xFFFFFF
				When <CMPEN> = 1 is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = 1 is set, an interrupt request (INTENCx) occurs. This bit has no effect on a value of <ZEN>.	

<INT[23:16]> is used only in Sensor mode (timer count) and Timer mode.

16.4.5 ENxCNT (Encoder Counter)

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

Bit	Bit Symbol	Type	Function		
31-24	−	R	Read as "0".		
23-0	CNT[23:0]	R/W	Encoder counter/capture value		
			Encoder mode:	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Encoder mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to the value of <RELOAD[15:0]>, it wraps around to 0 on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to 0, it is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.	
			Sensor mode: (event count)	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Sensor Event Count mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to 0xFFFF, it wraps around to 0 on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to 0, it wraps around to 0xFFFF on the next ENCLK.	
			Sensor mode: (Timer count)	Pulse detection time or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read. In Sensor mode, the value of encoder counter can be read and captured by software on each encoder pulse (ENCLK) by writing "1" to <SFTCAP>. The captured value is cleared to 0 by system reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Sensor Timer Count mode, the encoder counter is configured as a free-running counter that counts up with fsys. The encoder counter is cleared to 0 when the encoder pulse (ENCLK) is detected. When it has reached to 0xFFFFFF, it wraps around to 0 automatically.	
			Timer mode	Capture value of internal counter or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read and captured by software by writing "1" to <SFTCAP>.When <ZEN> = 1, the value of the encoder counter is also captured into <CNT[23:0]> on the Z edge selected by <ZESEL>. The captured value is cleared to "0" by reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached to 0xFFFFFF, it wraps around to 0 automatically.	

<CNT[23:16]> is used only in the sensor mode (Timer counting) or timer mode. In the encoder mode or sensor mode (event counting), always reads as "0".

16.5 Operational Description

16.5.1 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

16.5.2 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).

16.5.2.1 Event Count Mode

- Event detection (rotation pulse) → interrupt generation
- Event count → match interrupt occurs (measuring the amount of transfer)
- Rotation direction detection

16.5.2.2 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

16.5.3 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

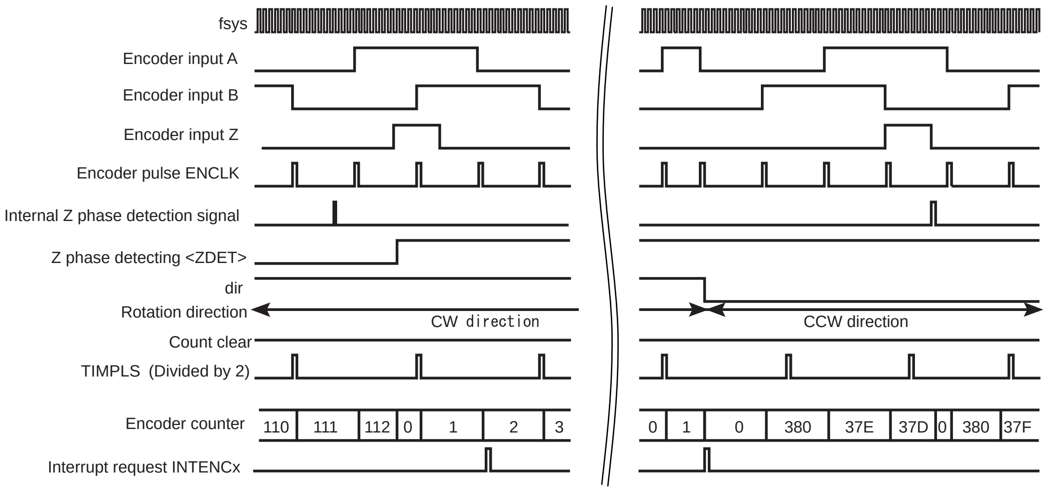
16.6 Function

16.6.1 Mode operation outline

16.6.1.1 Encoder mode

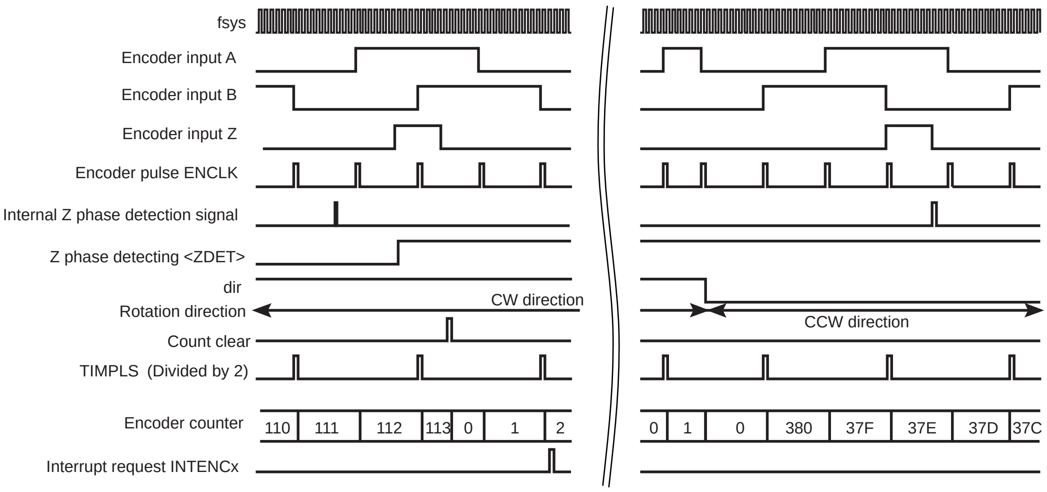
1. If ENxTNCR<ZEN> = 1

   (ENxRELOAD<RELOAD[15:0]> = 0x0380, ENxINT<INT[15:0]> = 0x0002)



2. If ENxTNCR<ZEN> = 0

   (ENxRELOAD<RELOAD[15:0]> = 0x0380, ENxINT<INT[15:0]> = 0x0002)

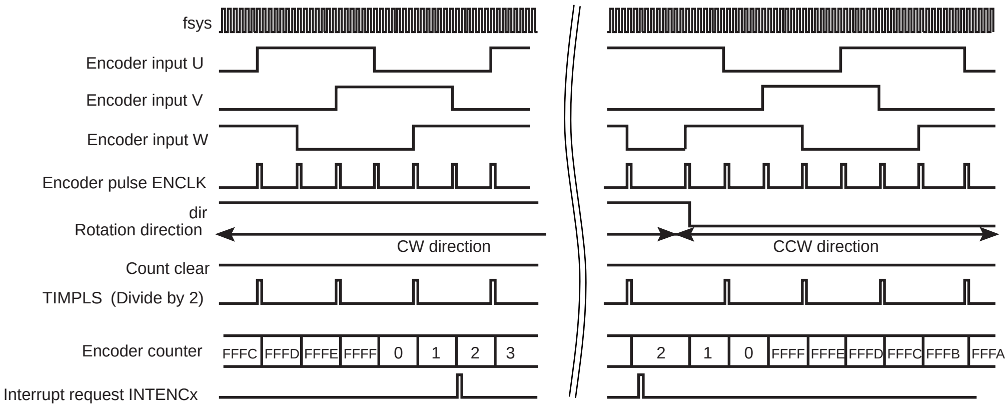


- The incremental encoder inputs of the MCU should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied by 4 clock derived from the decoded A and B quadrature signals.
- During CW rotation (i.e., A has the 90-degree phase lead to B), the encoder counter counts up; when it has reached to the value of ENxRELOAD<RELOAD[15:0]>, it wraps around to 0 on the next ENCLK.

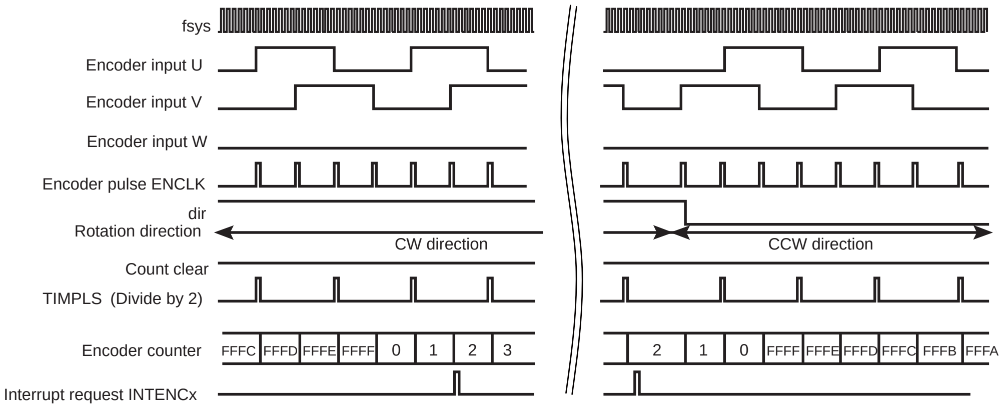
- During CCW rotation (i.e., A has the 90-degree phase lag to B), the encoder counter counts down; when it has reached to 0x0000, it is reloaded with the value of ENxRELOAD<RELOAD [15:0]> on the next ENCLK.
- Additionally, when ENxTNCR<ZEN> = 1, the encoder counter is cleared to 0 on the rising edge 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 increment or decrement.
- When ENxTNCR<ENCLR> is set to 1, the encoder counter is cleared to 0.
- ENxTNCR<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 ENxTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of ENxINT<INT[15:0]>. When ENxTNCR<ZEN> = 1, however, an interrupt does not occur while ENxTNCR<ZDET> = 0.
- When <ZDET> and <UD> are set to "0", ENxTNCR<ENRUN> is cleared to "0".

16.6.1.2 Sensor mode (event count)

1. If ENxTNCR<P3EN> = 1 (ENxINT<INT[15:0]> = 0x0002)



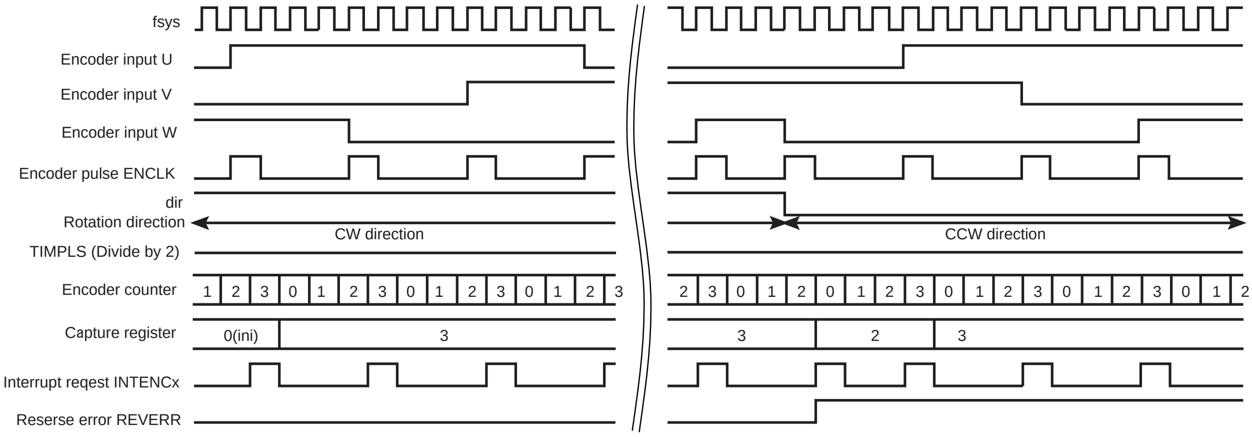
2. If ENxTNCR<P3EN> = 0 (ENxINT<INT[15:0]> = 0x0002)



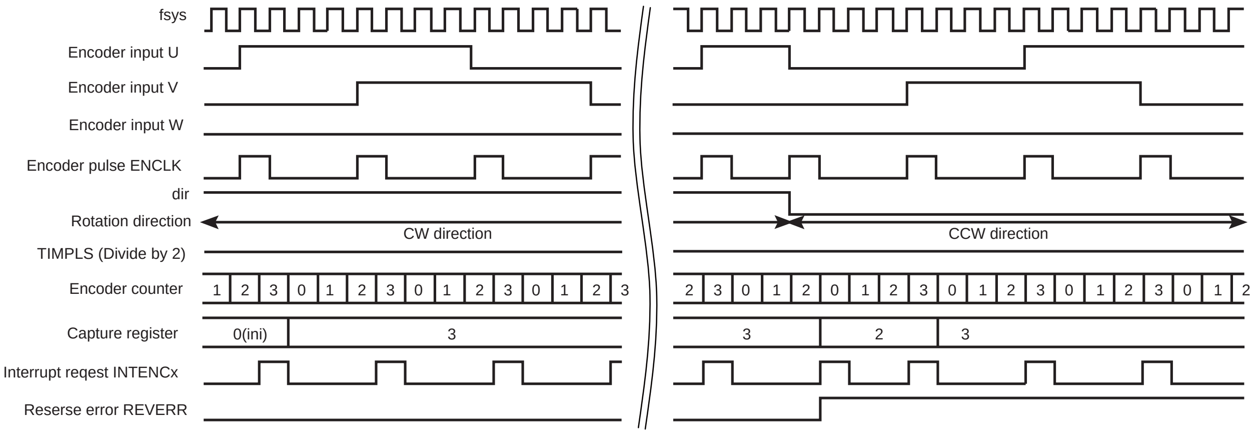
- The Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter counts the pulses of ENCLK, which is either multiplied by 4 clock (when ENxTNCR<P3EN> = 0) derived from the decoded U and V signals or multiplied by 6 clock (when ENxTNCR<P3EN> = 1) derived from the 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 ENxTNCR<ENCLR> is set to 1, the internal counter is cleared to 0.
- ENxTNCR<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 ENxTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the internal counter has reached to the value of ENxINT<INT[15:0]>.
- When ENxTNCR<UD> and ENxTNCR<ENRUN> are set to "0", <UD> is cleared to "0".

16.6.1.3 Sensor mode (Timer count)

1. If ENxTNCR<P3EN> = 1 (ENxINT<INT[23:0]> = 0x000002)



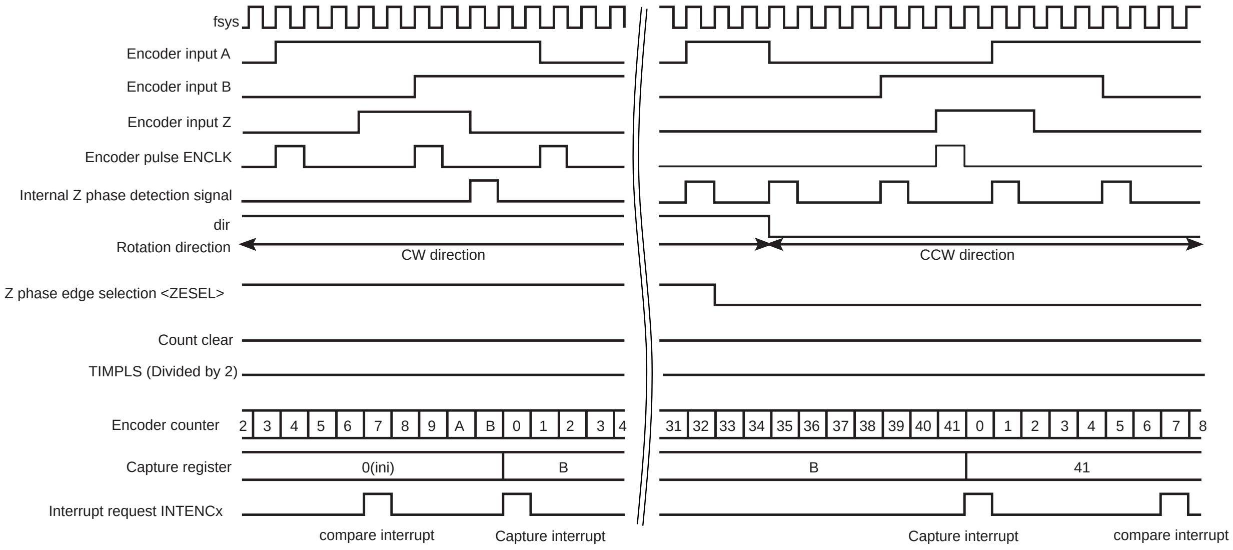
2. If ENxTNCR<P3EN> = 0 (ENxINT<INT[23:0]> = 0x000002)



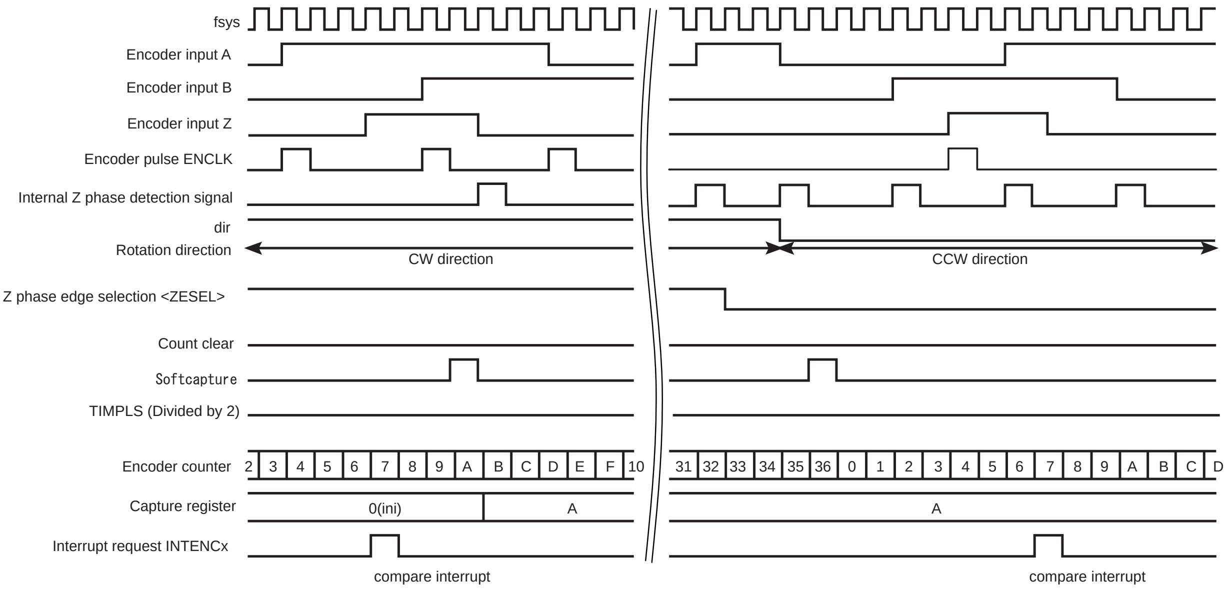
- In Sensor Timer Count mode, the Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied by 4 clock (when ENxTNCR<P3EN> = 0) derived from the decoded U and V signals or multiplied by 6 clock (when ENxTNCR<P3EN> = 1) derived from the decoded U, V and W signals.
- The encoder counter always counts up; it is cleared to 0 on ENCLK. When the encoder counter has reached to0xFFFFFF, it wraps around to 0.
- When ENxTNCR<ENCLR> is set to 1, the encoder counter is cleared to 0.
- ENCLK captures the value of the encoder counter into the ENxCNT register. The captured counter value can be read out of ENxCNT.
- Setting the software capture bit, ENxTNCR<SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENxCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENxCNT.
- ENxTNCR<UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- If ENxTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of ENxINT<INT[23:0]>.
- When ENxTNCR<ENRUN> is set to "0", ENxTNCR<UD> is cleared to "0".
- ENxTNCR<REVERR> is set to 1 when the rotation direction has changed. This bit is cleared to 0 on a read.
- The value of the ENxCNT register (the captured value) is retained, regardless of the value of ENxTNCR<ENRUN>. The ENxCNT register is only cleared by a reset.

16.6.1.4 Timer mode

1. If ENxTNCR<ZEN> = 1 (ENxINT<INT[23:0]> = 0x000006)



2. If ENxTNCR<ZEN> = 0 (ENxINT<INT[23:0]> = 0x000006)

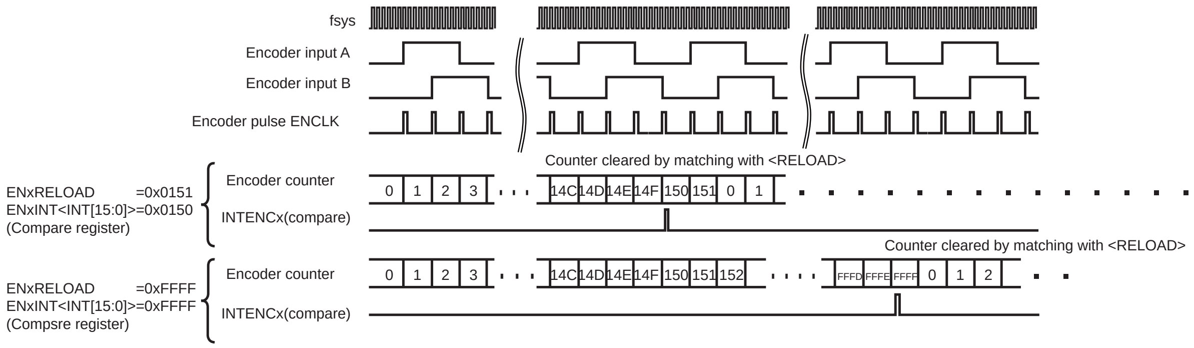


- When ENxTNCR<ZEN> = 1, the Z input pin is used as an external trigger. When ENxTNCR<ZEN> = 0, no external input is used to trigger the timer.
- The encoder counter always counts up. If ENxTNCR<ZEN> = 1, the counter is cleared to 0 on the rising edge of Z when ENxTNCR<ZESEL> is set to "0" and a falling edge when ENxTNCR<ZESEL> is set to "1". When the encoder counter has reached to 0xFFFFFF, it wraps around to 0.
- When ENxTNCR<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 ENxCNT register. The captured counter value can be read out of ENxCNT.
- Setting the software capture bit, ENxTNCR<SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENxCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENxCNT.
- ENxTNCR<UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- If ENxTNCR<CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of ENxINT<INT[23:0]>.

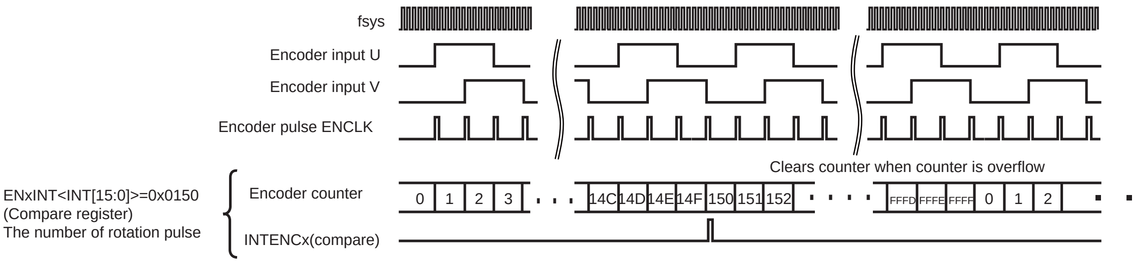
- When ENxTNCR<ENRUN> is set to "0", ENxTNCR<UD> is cleared to "0".
- The value of the ENxCNT register (the captured value) is retained, regardless of the value of ENxTNCR<ENRUN>. The ENxCNT register is only cleared by a reset.

16.6.2 Counter and interrupt generate operation when ENxTNCR<CMPEN> = 1

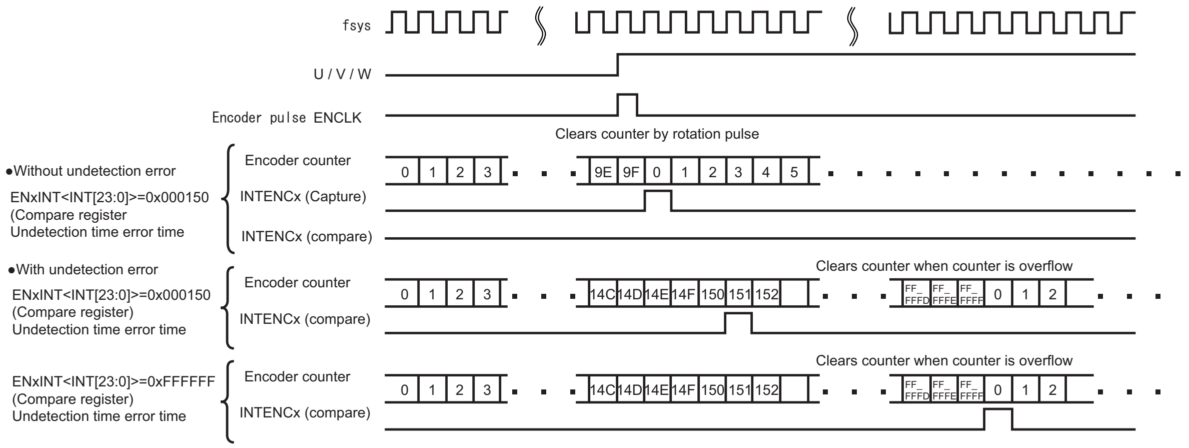
16.6.2.1 Encoder mode



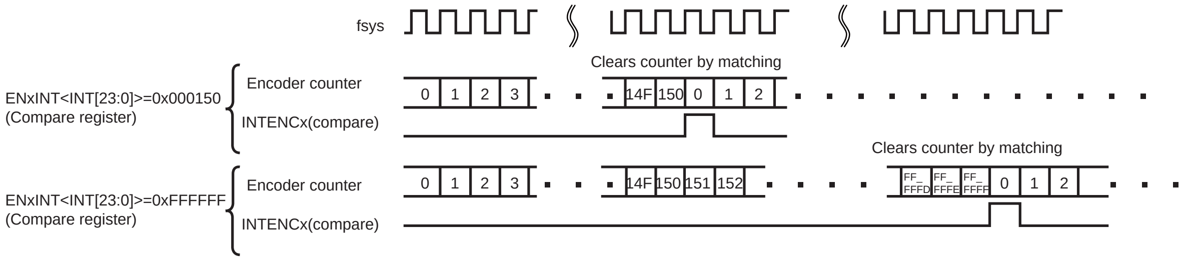
16.6.2.2 Sensor mode (event count)



16.6.2.3 Sensor mode (Timer count)


fsys
U / V / W
Encoder pulse ENCLK
Clears counter by rotation pulse
●Without undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register
Undetection time error time
Encoder counter
0 1 2 3 9E 9F 0 1 2 3 4 5
INTENCx (Capture)
INTENCx (compare)
●With undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register)
Undetection time error time
Clears counter when counter is overflow
Encoder counter
0 1 2 3 14C 14D 14E 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2
INTENCx (compare)
ENxINT<INT[23:0]>=0xFFFFFF
(Compare register)
Undetection time error time
Clears counter when counter is overflow
Encoder counter
0 1 2 3 14C 14D 14E 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2
INTENCx (compare)

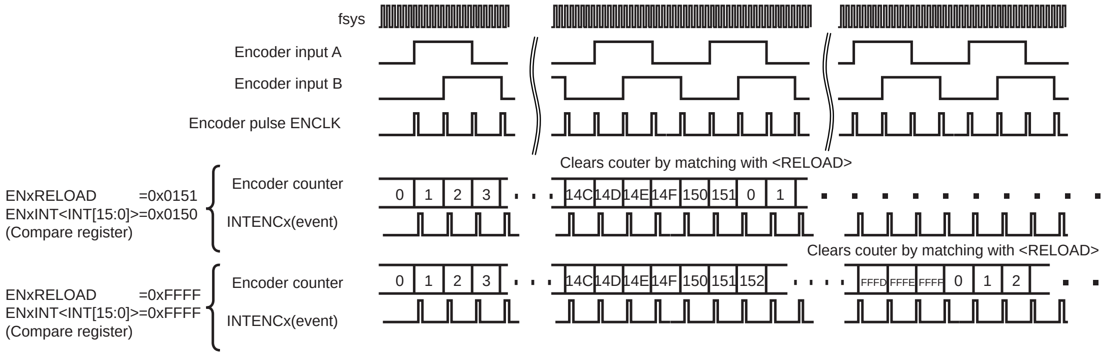
16.6.2.4 Timer mode



16.6.3 Counter and interrupt generate operation when ENxTNCR<CMPEN> = 0

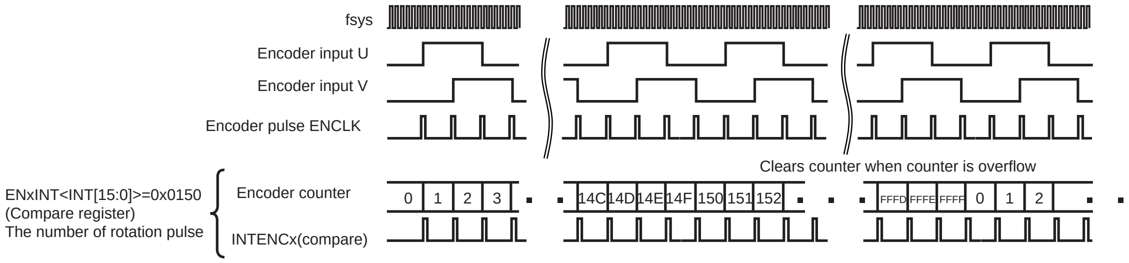
16.6.3.1 Encoder mode

ENxTNCR<ENDEV[2:0]>="000"

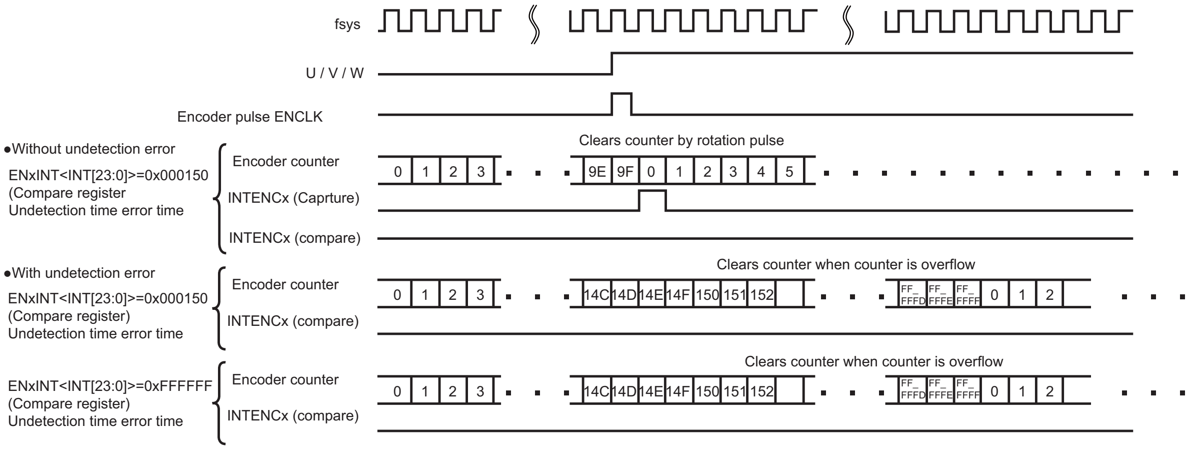


16.6.3.2 Sensor mode (event count)

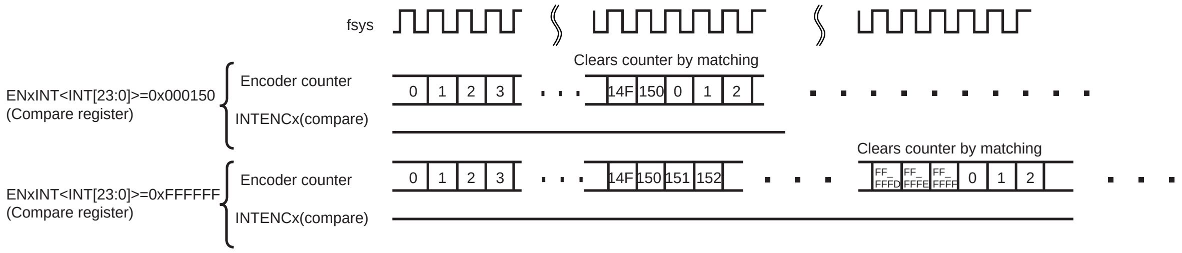
ENxTNCR<ENDEV[2:0]>="000"



16.6.3.3 Sensor mode (Timer count)


fsys
U / V / W
Encoder pulse ENCLK
●Without undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register
Undetection time error time
Clears counter by rotation pulse
Encoder counter
INTENCx (Caprture)
INTENCx (compare)
0 1 2 3 9E 9F 0 1 2 3 4 5
●With undetection error
ENxINT<INT[23:0]>=0x000150
(Compare register)
Undetection time error time
Clears counter when counter is overflow
Encoder counter
INTENCx (compare)
0 1 2 3 14C 14D 14E 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2
ENxINT<INT[23:0]>=0xFFFFFF
(Compare register)
Undetection time error time
Clears counter when counter is overflow
Encoder counter
INTENCx (compare)
0 1 2 3 14C 14D 14E 14F 150 151 152 FF_FFFD FF_FFFE FF_FFFF 0 1 2

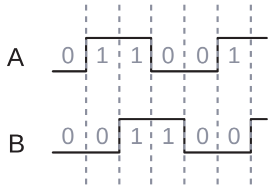
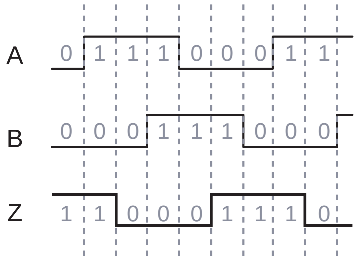
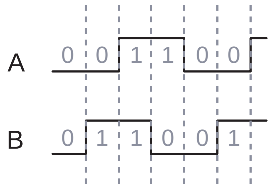
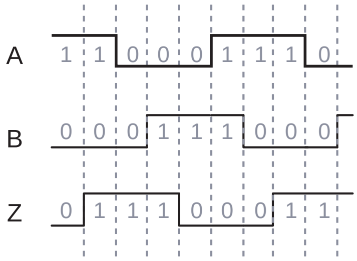
16.6.3.4 Timer mode



16.6.4 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 ENxTNCR<P3EN> = 1.

	2-phase input	3-phase input
CW direction		 A 0 1 1 1 0 0 0 1 1 B 0 0 0 1 1 1 0 0 0 Z 1 1 0 0 0 1 1 1 0
CCW direction		 A 1 1 0 0 0 1 1 1 0 B 0 0 0 1 1 1 0 0 0 Z 0 1 1 1 0 0 0 1 1

16.6.5 Counter Circuit

The counter circuit has a 24-bit up/down counter.

16.6.5.1 Operation Description

Depending on the operation modes, counting, clearing and reloading operation are controlled as described in Table 16-2.

Table 16-2 Counter control

Mode <MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Count	Operation	Counter clear condition	Counter reload condition	Operational range of counter (Reload value)
Encoder mode 00	0	0	A,B	Encoder pulse (ENCLK)	UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD>	-	0x0000 to <RELOAD>
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
	1		A,B,Z		UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD> [3] Z-trigger	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (event count) 01	0	0	U,V		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	0x0000 to 0xFFFF
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
		1	U,V,W		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (Timer count) 10	0	0	U,V	fsys	UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Encoder pulse (ENCLK)	-	0x000000 to 0xFFFFFF
		1	U,V,W		UP		-	
Timer mode 11	0	×	-		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <INT[23:0]>	-	0x000000 to 0xFFFFFF
	1		Z		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <INT[23:0]> [4] Z-trigger	-	

Note: The counter value is not cleared by writing "0" to ENxTNCR<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.

16.6.6 Interrupt

The interrupt consists of four interrupts including Event (divide pulse and capture), Abnormal detecting time, Timer compare and Capture interrupts.

16.6.6.1 Operational Description

When ENxTNCR<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 ENxTNCR<CMPEN> and <ZEN>.Table 16-3 shows interrupt factors.

Table 16-3 Interrupt factors

	Interrupt factor	Description	Mode	Interrupt output	Status flag
1	Event count interrupt	When <CMPEN> = 1, the encoder counter counts events (encoder pulses). When it has reached to the value programmed in <INT[15:0]>, an interrupt occurs.	Encoder mode and Sensor mode (event count)	<INTEN> = 1 and <CMPEN> = 1	<CMP>
2	Event interrupt (divide pulse)	An interrupt occurs on each divided clock pulse (1 to 128 divide), which is derived by dividing the encoder pulse by a factor programmed in <ENDEV>.		<INTEN> = 1	Not available
3	Event interrupt (capture interrupt)	An interrupt occurs to indicate that an event (encoder pulse) has occurred, causing the counter value to be captured on the rotation pulse timing.	Sensor mode (Timer count)	<INTEN> = 1	Not available
4	Abnormal detection time error interrupt	When <CMPEN> = 1, the ENC uses a counter that counts up with fsys and is cleared by an event (encoder pulse). If no event occurs for a period of time programmed in <INT[23:0]>, an interrupt occurs.		<INTEN> = 1 and <CMPEN> = 1	<CMP>
5	Timer compare interrupt	When <CMPEN> = 1, an interrupt occurs when the timer has reached to the value programmed in <INT[23:0]>.	Timer mode	<INTEN> = 1 and <CMPEN> = 1	<CMP>
6	Capture interrupt	An interrupt occurs when the counter value has been captured on an external trigger (Z input).		<INTEN> = 1	Not available

In Sensor Timer Count mode and Timer mode, the value of the encoder counter can be captured into the ENxCNT register.

The captured counter value can be read out of the ENxCNT register.

In Sensor Timer Count mode, the value of the encoder counter is captured into the ENxCNT register upon occurrence of an event (encoder pulse). The counter value can also be captured by writing a 1 to ENxTNCR<SFTCAP> by software.

In Timer mode, the counter value can be captured by writing a 1 to ENxTNCR<SFTCAP> by software. If ENxTNCR<ZEN> is set to 1, the counter value can also be captured by an edge of the Z signal input selected according to ENxTNCR<ZESEL> by external trigger.

17. Real Time Clock (RTC)

17.1 Function

1. Clock (hour, minute and second)
2. Calendar (month, week, date and leap year)
3. Selectable 12 (am/ pm) and 24 hour display
4. Time adjustment + or − 30 seconds (by software)
5. Alarm (alarm output)
6. Alarm interrupt

17.2 Block Diagram

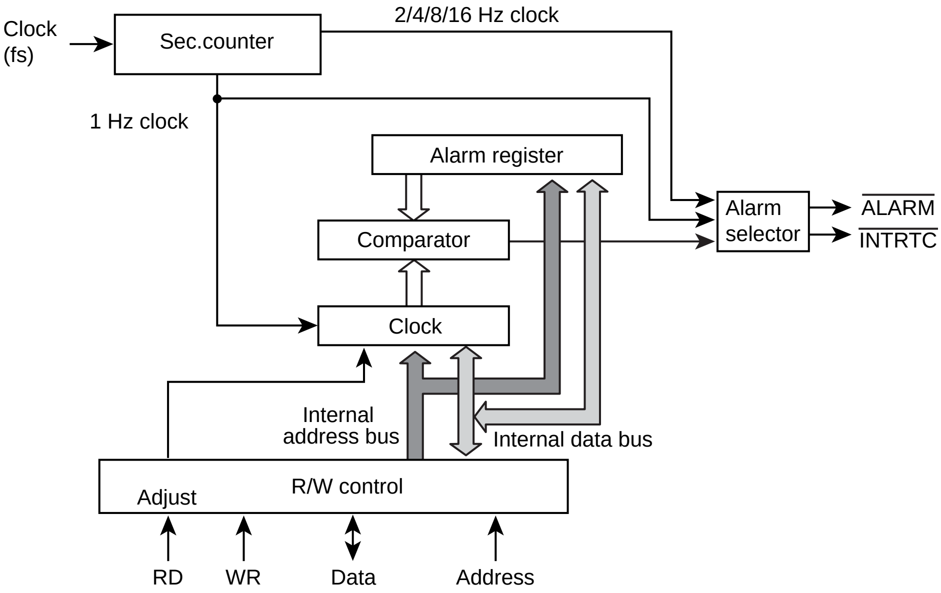


Figure 17-1 Block Diagram

Note 1: Western calendar year column: This product uses only the final two digits of the year. The year following 99 is 00 years. Please take into account the first two digits when handling years in the western calendar.

Note 2: Leap year: A leap year is divisible by 4 excluding a year divisible by 100; the year divisible by 100 is not considered to be a leap year. Any year divisible by 400 is a leap year. This product is considered the year divisible by 4 to be a leap year and does not take into account the above exceptions. It needs adjustments for the exceptions.

17.3 Detailed Description Register

17.3.1 Register List

The registers and the addresses related to RTC are shown as below.

RTC has two functions, PAGE0 (clock) and PAGE1 (alarm), which share some parts of registers.

The PAGE can be selected by setting RTCPAGER<PAGE >.

Base Address = 0x4004_0100

Register name		Address (Base+)
Second column register (only PAGE0)	RTCSECR	0x0000
Minute column register	RTCMINR	0x0001
Hour column register	RTCHOURR	0x0002
- (note 1)	-	0x0003
Day of the week column register	RTCDAYR	0x0004
Day column register	RTCDATER	0x0005
Month column register (PAGE0) Selection register of 24-hour,12-hour (PAGE1)	RTCMONTHR	0x0006
Year column register (PAGE0) Leap year register (PAGE1)	RTCYEARR	0x0007
PAGE register	RTCPAGER	0x0008
- (note 1)	-	0x0009
- (note 1)	-	0x000A
- (note 1)	-	0x000B
Reset register	RTCRESTR	0x000C
Reserved	-	0x000D
- (note 1)	-	0x000E
- (note 1)	-	0x000F

Note 1: "0" is read by reading the address. Writing is disregarded.
Note 2: Access to the "Reserved" areas is prohibited.

17.3.2 Control Register

Reset operation initializes the following registers.

- RTCPAGER<PAGE>, <ADJUST>, <INTENA>
- RTCRESTR<RSTALM>, <RSTTMR>, <DIS16HZ>, <DIS1HZ>, <DIS2HZ>, <DIS4HZ>, <DIS8HZ>

Other clock-related registers are not initialized by reset operation.

Before using the RTC, set the time, month, day, day of the week, year and leap year in the relevant registers.

Caution is required in setting clock data, adjusting seconds or resetting the clock.

Refer to "17.4.3 Entering the Low Power Consumption Mode" for more information.

Note: In this product, external oscillation connect pins (XT1/XT2) are dual-purpose ports with general-purpose ports. If these pins are used as an external oscillation connect pin, refer to the setting procedure in Chapter Clock/mode Control.

Table 17-1 PAGE0 (clock function) register

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RTCSECR	–	40s	20s	10s	8s	4s	2s	1s	Second column
RTCMINR	–	40min	20min	10min	8min	4min	2min	1min	Minute column
RTCHOURR	–	–	20hours PM/AM	10hours	8hours	4hours	2hours	1hour	Hour column
RTCDAYR	–	–	–	–	–	Day of the week			Day of the week column
RTCDATER	–	–	Day20	Day10	Day8	Day4	Day2	Day1	Day column
RTCMONTHR	–	–	–	Oct.	Aug.	Apr.	Feb.	Jan.	Month column
RTCYEARR	year 80	year 40	year20	year 10	year 8	year 4	year 2	year 1	Year column (lower two columns)
RTCPAGER	Interrupt enable	–	–	Adjustment function	Clock enable	Alarm enable	–	PAGE setting	PAGE register
RTCRESTR	1 Hz enable	16 Hz enable	Clock reset	Alarm reset	–	2Hz enable	4Hz enable	8Hz enable	Reset register

Note: Reading RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE0 captures the current state.

Table 17-2 PAGE1 (alarm function) registers

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function
RTCSECR	–	–	–	–	–	–	–	–	–
RTCMINR	–	40min	20min	10min	8min	4min	2min	1min	Minute column
RTCHOURR	–	–	20hours PM/AM	10hours	8hours	4hours	2hours	1hour	Hour column
RTCDAYR	–	–	–	–	–	Day of the week			Day of the week column
RTCDATER	–	–	Day20	Day10	Day8	Day4	Day2	Day1	Day column
RTCMONTHR	–	–	–	–	–	–	–	24/12	24-hour clock mode
RTCYEARR	–	–	–	–	–	–	Leap-year setting		Leap-year mode
RTCPAGER	Interrupt enable	–	–	Adjustment function	Clock enable	Alarm enable	–	PAGE setting	PAGE register
RTCRESTR	1 Hz Enable	16 Hz Enable	Clock reset	Alarm reset	–	2Hz enable	4Hz enable	8Hz enable	Reset register

Note 1: Reading RTCMINR, RTCHOURR, RTCDAYR, RTCMONTHR, RTCYEARR of PAGE1 captures the current state.

Note 2: RTCSECR, RTCMINR, RTCHOURR, RTCDAYR, RTCDATER, RTCMONTHR, RTCYEARR of PAGE0 and RTCYEARR of PAGE1 (for leap year) must be read twice and compare the data captured.

17.3.3 Detailed Description of Control Register

17.3.3.1 RTCSECR (Second column register (for PAGE0 only))

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

Bit	Bit Symbol	Type	Function
7	−	R	Read as 0.
6-0	SE	R/W	Setting digit register of second 000_0000 : 00sec. 001_0000 : 10sec. 010_0000 : 20sec. 000_0001 : 01sec. 001_0001 : 11sec. · 000_0010 : 02sec. 001_0010 : 12sec. 011_0000 : 30sec. 000_0011 : 03sec. 001_0011 : 13sec. · 000_0100 : 04sec. 001_0100 : 14sec. 100_0000 : 40sec. 000_0101 : 05sec. 001_0101 : 15sec. · 000_0110 : 06sec. 001_0110 : 16sec. 101_0000 : 50sec. 000_0111 : 07sec. 001_0111 : 17sec. · 000_1000 : 08sec. 001_1000 : 18sec. · 000_1001 : 09sec. 001_1001 : 19sec. 101_1001 : 59sec.

Note: The setting other than listed above is prohibited.

17.3.3.2 RTCMINR (Minute column register (PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	MI						
After reset	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Functon
7	−	R	Read as 0.
6-0	MI	R/W	Setting digit register of Minutes. 000_0000 : 00min. 001_0000 : 10min. 010_0000 : 20min. 111_1111 : don't care 000_0001 : 01min. 001_0001 : 11min. · (Only PAGE1) 000_0010 : 02min. 001_0010 : 12min. 011_0000 : 30min. 000_0011 : 03min. 001_0011 : 13min. · 000_0100 : 04min. 001_0100 : 14min. 100_0000 : 40min. 000_0101 : 05min. 001_0101 : 15min. · 000_0110 : 06min. 001_0110 : 16min. 101_0000 : 50min. 000_0111 : 07min. 001_0111 : 17min. · 000_1000 : 08min. 001_1000 : 18min. · 000_1001 : 09min. 001_1001 : 19min. 101_1001 : 59min.

Note: The setting other than listed above is prohibited.

17.3.3.3 RTCHOURR (Hour column register(PAGE0/1))

(1) 24-hour clock mode (RTCMONTHR<MO0>= "1")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-6	–	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. 00_0000 : 0 o'clock 00_0001 : 1 o'clock 00_0010 : 2 o'clock 00_0011 : 3 o'clock 00_0100 : 4 o'clock 00_0101 : 5 o'clock 00_0110 : 6 o'clock 00_0111 : 7 o'clock 00_1000 : 8 o'clock 00_1001 : 9 o'clock 01_0000 : 10 o'clock 01_0001 : 11 o'clock 01_0010 : 12 o'clock 01_0011 : 13 o'clock 01_0100 : 14 o'clock 01_0101 : 15 o'clock 01_0110 : 16 o'clock 01_0111 : 17 o'clock 01_1000 : 18 o'clock 01_1001 : 19 o'clock 10_0000 : 20 o'clock 10_0001 : 21 o'clock 10_0010 : 22 o'clock 10_0011 : 23 o'clock 11_1111 : don't care (Only PAGE1)

Note: The setting other than listed above is prohibited.

(2) 12-hour clock mode (RTCMONTHR<MO0> = "0")

	7	6	5	4	3	2	1	0
Bit symbol	-	-	HO					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-6	–	R	Read as 0.
5-0	HO	R/W	Setting digit register of Hour. (AM) 00_0000 : 0 o'clock 00_0001 : 1 o'clock 00_0010 : 2 o'clock 00_0011 : 3 o'clock 00_0100 : 4 o'clock 00_0101 : 5 o'clock 00_0110 : 6 o'clock 00_0111 : 7 o'clock 00_1000 : 8 o'clock 00_1001 : 9 o'clock 01_0000 : 10 o'clock 01_0001 : 11 o'clock (PM) 10_0000 : 0 o'clock 10_0001 : 1 o'clock 10_0010 : 2 o'clock 10_0011 : 3 o'clock 10_0100 : 4 o'clock 10_0101 : 5 o'clock 10_0110 : 6 o'clock 10_0111 : 7 o'clock 10_1000 : 8 o'clock 10_1001 : 9 o'clock 11_0000 : 10 o'clock 11_0001 : 11 o'clock 11_1111 : don't care (Only PAGE1)

Note: The setting other than listed above is prohibited.

17.3.3.4 RTCDAYR (Day of the week column register(PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	WE		
After reset	0	0	0	0	0	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-3	−	R	Read as 0.
2-0	WE	R/W	Setting digit register of day of the week. 000: Sunday 111: don’t care (Only PAGE1) 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday

Note: The setting other than listed above is prohibited.

17.3.3.5 RTCDATER (Day column register (for PAGE0/1 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	DA					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-6	−	R	Read as 0.
5-0	DA	R/W	Setting digit register of day. 01_0000 : 10th day 10_0000 : 20th day 11_0000 : 30th day 00_0001 : 1st day 01_0001 : 11th day 10_0001 : 21th day 11_0001 : 31th day 00_0010 : 2nd day 01_0010 : 12th day 10_0010 : 22th day 00_0011 : 3rd day 01_0011 : 13th day 10_0011 : 23th day 11_1111 : don’t care (Only PAGE1) 00_0100 : 4th day 01_0100 : 14th day 10_0100 : 24th day 00_0101 : 5th day 01_0101 : 15th day 10_0101 : 25th day 00_0110 : 6th day 01_0110 : 16th day 10_0110 : 26th day 00_0111 : 7th day 01_0111 : 17th day 10_0111 : 27th day 00_1000 : 8th day 01_1000 : 18th day 10_1000 : 28th day 00_1001 : 9th day 01_1001 : 19th day 10_1001 : 29th day

Note 1: The setting other than listed above is prohibited.

Note 2: Do not set for non-existent days (e.g. 30th Feb.).

17.3.3.6 RTCMONTHR (Month column register (for PAGE0 only))

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	MO				
After reset	0	0	0	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit Symbol	Type	Function
7-5	−	R	Read as 0.
4-0	MO	R/W	Setting digit register of Month. 0_0001 : January 0_0111 : July 0_0010 : February 0_1000 : August 0_0011 : March 0_1001 : September 0_0100 : April 1_0000 : October 0_0101 : May 1_0001 : November 0_0110 : June 1_0010 : December

Note: The setting other than listed above is prohibited.

17.3.3.7 RTCMONTHR (Selection of 24-hour clock or 12-hour clock (for PAGE1 only))

	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	MO0
After reset	0	0	0	0	0	0	0	Undefined

Bit	Bit Symbol	Type	Function
7-1	−	R	Read as 0.
0	MO0	R/W	0: 12-hour 1: 24-hour

Note: Do not change the RTCMONTHR<MO0> while the RTC is in operation.

17.3.3.8 RTCYEARR (Year column register (for PAGE0 only))

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

Bit	Bit Symbol	Type	Function		
7-0	YE	R/W	Setting digit register of Year.		
			0000_0000 : 00 year	0001_0000 : 10 years	0110_0000 : 60 years
			0000_0001 : 01 years	·	·
			0000_0010 : 02 years	0010_0000 : 20 years	0111_0000 : 70 years
			0000_0011 : 03 years	·	·
			0000_0100 : 04 years	0011_0000 : 30 years	1000_0000 : 80 years
			0000_0101 : 05 years	·	·
			0000_0110 : 06 years	0100_0000 : 40 years	1001_0000 : 90 years
			0000_0111 : 07 years	·	·
			0000_1000 : 08 years	01001_0000 : 50 years	·
			0000_1001 : 09 years	·	1001_1001 : 99 years

Note: The setting other than listed above is prohibited.

17.3.3.9 RTCYEARR (Leap year register (for PAGE1 only))

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

Bit	Bit Symbol	Type	Function
7-2	−	R	Read as 0.
1-0	LEAP	R/W	00 : A leap year 01 : one year after a leap year 10 : two years after a leap year 11 : three years after a leap year

17.3.3.10 RTCPAGER(PAGE register(PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	INTENA	-	-	ADJUST	ENATMR	ENAALM	-	PAGE
After reset	0	0	0	0	Undefined	Undefined	0	0

Bit	Bit Symbol	Type	Function
7	INTENA	R/W	INTRTC 0:Disable 1:Enable
6-5	−	R	Read as 0.
4	ADJUST	R/W	[Write] 0: Don't care 1: Sets ADJUST request Adjusts seconds. The request is sampled when the sec. counter counts up. If the time elapsed is between 0 and 29 seconds, the sec. counter is cleared to "0". If the time elapsed is between 30 and 59 seconds, the min. counter is carried and sec. counter is cleared to "0". [Read] 0: ADJUST no request 1: ADJUST requested If "1" is read, it indicates that ADJUST is being executed. If "0" is read, it indicates that the execution is finished.
3	ENATMR	R/W	Clock 0: Disable 1: Enable
2	ENAALM	R/W	ALARM 0: Disable 1: Enable
1	−	R	Read as 0.
0	PAGE	R/W	PAGE selection 0:Selects Page0 1:Selects Page1

Note 1: A read-modify-write operation cannot be performed.

Note 2: To set interrupt enable bits to <ENATMR>, <ENAALM> and <INTENA>, you must follow the order specified here. Make sure not to set them at the same time (make sure that there is time lag between interrupt enable and clock/ alarm enable).To change the setting of <ENATMR> and <ENAALM>, <INTENA> must be disabled first.

Example: Clock setting/Alarm setting

		7	6	5	4	3	2	1	0	
RTCPAGER	←	0	0	0	0	1	1	0	0	Enables Clock and alarm
RTCPAGER	←	1	0	0	0	1	1	0	0	Enables interrupt

17.3.3.11 RTCRESTR (Reset register (for PAGE0/1))

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	-	DIS2HZ	DIS4HZ	DIS8HZ
After reset	1	1	0	0	0	1	1	1

Bit	Bit Symbol	Type	Function
7	DIS1HZ	R/W	1 Hz 0:Enable 1: Disable
6	DIS16HZ	R/W	16 Hz 0: Enable 1: Disable
5	RSTTMR	R/W	[Write] 0: Don't care 1: Sec.counter reset Resets the sec counter. The equest is sampled using low-speed clock. [Read] 0: No reset request 1: RESET requested If "1" is read, it indicates that RESET is being executed. If "0" is read, it indicates that the execution is finished.
4	RSTALM	R/W	0:Don't care 1: Alarm reset Initializes alarm registers (Minute column, hour column, day column and day of the week column) as follows. MInute:00, Hour:00, Day:01, Day of the week:Sunday
3	−	R	Read as 0.
2	DIS2HZ	R/W	2 Hz 0: Enable 1: Disable
1	DIS4HZ	R/W	4 Hz 0: Enable 1: Disable
0	DIS8HZ	R/W	8 Hz 0: Enable 1: Disable

Note: A read-modify-write operation cannot be performed.

The setting of <DIS1HZ>, <DIS2HZ>, <DIS4HZ>, <DIS8HZ>, <DIS16HZ> and RTCPAGER<ENAALM> used for alarm, 1Hz interrupt, 2Hz interrupt, 4Hz interrupt, 8Hz interrupt and 16Hz interrupt are shown as below.

Table 17-3 Select interrupt source signal

<DIS1HZ>	<DIS2HZ>	<DIS4HZ>	<DIS8HZ>	<DIS16HZ>	RTCPAGER <ENAALM>	Interrupt source signal
1	1	1	1	1	1	ALARM
0	1	1	1	1	0	1 Hz
1	0	1	1	1	0	2 Hz
1	1	0	1	1	0	4Hz
1	1	1	0	1	0	8Hz
1	1	1	1	0	0	16 Hz
Others						Interrupt not generated

17.4 Operational Description

The RTC incorporates a second counter that generates a 1Hz signal from a 32.768 kHz signal.

The second counter operation must be taken into account when using the RTC.

Note: After reset, a low-speed clock stops oscillation. XT1 and XT2 ports are initialized to PP0 and PP1 ports. Re-setup RTC register.

17.4.1 Reading clock data

1. Using 1Hz interrupt

   The 1Hz interrupt is generated being synchronized with counting up of the second counter.

   Data can be read correctly if reading data after 1Hz interrupt occurred.

2. Using pair reading

   There is a possibility that the clock data may be read incorrectly if the internal counter operates carry during reading. To ensure correct data reading, read the clock data twice as shown below. A pair of data read successively needs to match.

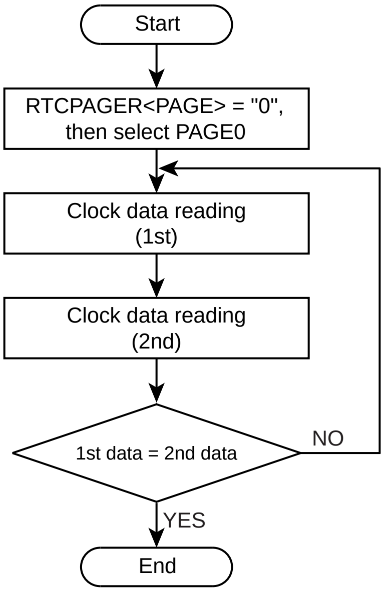


Figure 17-2 Flowchart of the clock data reading

17.4.2 Writing clock data

A carry during writing ruins correct data writing. The following procedure ensures the correct data writing.

1. Using 1 Hz interrupt

   The 1Hz interrupt is generated by being synchronized with counting up of the second counter. If data is written in the time between 1Hz interrupt and subsequent one second count, it completes correctly.

2. Resetting counter

Write data after resetting the second counter.

The 1Hz-interrupt is generated one second after enabling the interrupt subsequent to counter reset.

The time must be set within one second after the interrupt.

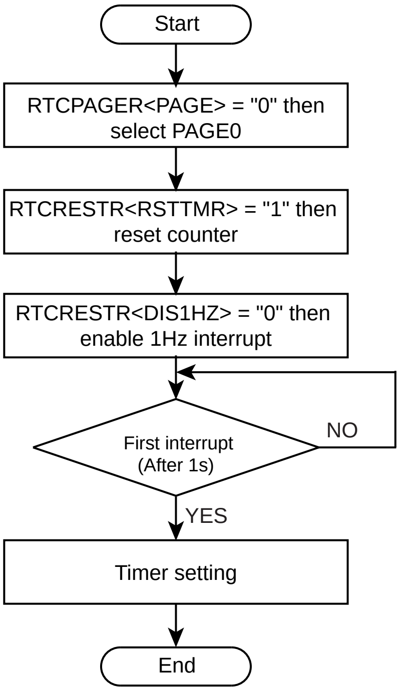

Start
RTCPAGER<PAGE> = "0" then select PAGE0
RTCRESTR<RSTTMR> = "1" then reset counter
RTCRESTR<DIS1HZ> = "0" then enable 1Hz interrupt
First interrupt (After 1s)
NO
YES
Timer setting
End

Figure 17-3 Flowchart of the clock data writing

3. Disabling the clock

Writing "0" to RTCPAGER<ENATMR> disables clock operation including a carry.

Stop the clock after the 1Hz-interrupt. The second counter keeps counting.

Set the clock again and enable the clock within one second before next 1Hz-interrupt

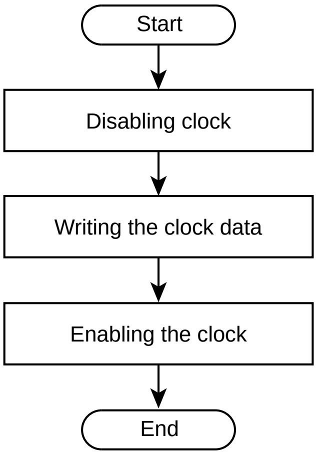


Figure 17-4 Flowchart of the disabling clock

17.4.3 Entering the Low Power Consumption Mode

To enter SLEEP mode, in which the system clock stops, after changing clock data, adjusting seconds or resetting the clock, be sure to observe one of the following procedures

1. After changing the clock setting registers, setting the RTCPAGER<ADJUST> bit or setting the RTCRESTR<RSTTMR> bit, wait for one second for an interrupt to be generated.

2. After changing the clock setting registers, setting the RTCPAGER<ADJUST> bit or setting the RTCRESTR<RSTTMR> bit, read the corresponding clock register values, <ADJUST> or <RSTTMR> to make sure that the setting you have made is reflected.

17.5 Alarm function

By writing "1" to RTCPAGER<PAGE>, the alarm function of the PAGE1 registers is enabled. One of the following three signals is output to the ALARM pin.

1. "Low" pulse (when the alarm register corresponds with the clock)

2. Outputting Low-pulse (1 Hz, 2 Hz, 4 Hz, 8 Hz or 16 Hz)

In any cases shown above, the INTRTC outputs one cycle pulse of low-speed clock. It outputs the INTRTC interrupt request simultaneously.

The INTRTC interrupt signal is falling edge triggered. Specify the falling edge as the active state in the CG Interrupt Mode Control Register

17.5.1 "Low" pulse (when the alarm register corresponds with the clock)

"Low" pulse is output to the ALARM pin when the values of the PAGE0 clock register and the PAGE1 alarm register correspond. The INTRTC interrupt is generated and the alarm is triggered.

The alarm settings

Initialize the alarm with alarm prohibited. Write "1" to RTCRESTR<RSTALM>.

It makes the alarm setting to be 00 minute, 00 hour, 01 day and Sunday.

Setting alarm for min., hour, date and day is done by writing data to the relevant PAGE1 register.

Enable the alarm with the RTCPAGER <ENAALM> bit. Enable the interrupt with the RTCPAGER <INTENA> bit.

The following is an example program for outputting an alarm from the ALARM pin at noon (12:00) on Monday 5th.

		7	6	5	4	3	2	1	0	
RTCPAGER	←	0	0	0	0	1	0	0	1	Disables alarm, sets PAGE1
RTCRESTR	←	1	1	0	1	0	0	0	0	Initializes alarm
RTCDAYR	←	0	0	0	0	0	0	0	1	Monday
RTCDATER	←	0	0	0	0	0	1	0	1	5th day
RTCHOURR	←	0	0	0	1	0	0	1	0	Sets 12 o’clock
RTCMINR	←	0	0	0	0	0	0	0	0	Sets 00 min
RTCPAGER	←	0	0	0	0	1	1	0	0	Enables alarm
RTCPAGER	←	1	0	0	0	1	1	0	0	Enables interrupts

The above alarm works in synchronization with the low-speed clock. When the CPU is operating at high frequency oscillation, a maximum of one clock delay at fs (about 30μs) may occur for the time register setting to become valid.

17.5.2 Outputting Low-pulse (1 Hz, 2 Hz, 4 Hz, 8 Hz or 16 Hz)

When RTCPAGER<ENAALM> and RTCRESTR are set as Table 17-3 and then RTCPAGER<INTENA> = "1" is set, one cycle low-speed (1 Hz, 2 Hz, 4 Hz, 8 Hz or 16 Hz) low pulse is output to $\overline{\text{ALARM}}$ pin. At the same time, INTRTC interrupt is also output.

18. Power-on-Reset Circuit (POR)

The power-on-reset circuit (POR) generates a power-on reset signal when power-on.

Power supply voltage is indicated as DVDD5.

18.1 Structure

Power-on-reset circuit consists of the reference voltage generation circuit, comparators and the power-on counter.

This circuit compares a voltage divided by the ladder resistor with a reference voltage generated in the reference voltage generation circuit in the comparator.

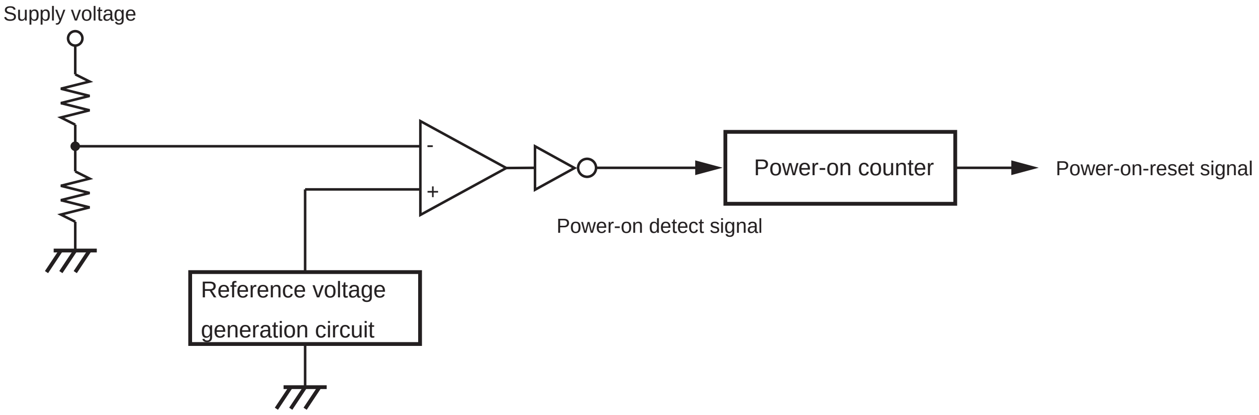


Figure 18-1 Power-on-reset circuit

18.2 Function

At power-on, a power-on detection signal generates while power supply voltage is lower than the releasing voltage. Power-on detection signal is released at the timing when DVDD5 is over 2.8 V.

If the power-on detection signal is released and the reset detection signal is also released, the power-on counter starts to operate. After waiting time (approximately 0.8ms) has elapsed, the power-on reset signal is released.

During the power-on reset signal generation, the CPU and the peripheral functions are reset.

When a reset input is not used, supply voltage must be raised to the recommended operational voltage range until the power-on reset releasing. If power supply voltage does not reach to the recommended operational voltage range during this period, TMPM3U6FY/FW cannot operate properly.

Note: Due to the fluctuation of supply voltage, the power-on reset circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

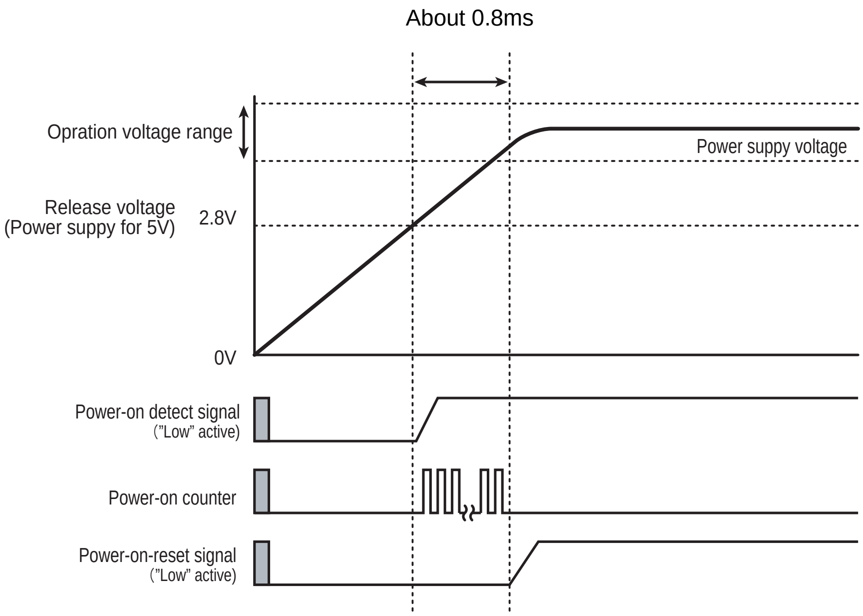


Figure 18-2 Power-on-reset operation timing

19. Voltage Detection Circuit (VLTD)

Voltage detection circuit generates an interrupt signal by detecting a decreasing voltage.

Supply voltage is indicated as DVDD5.

Note: Due to the fluctuation of supply voltage, the power-on reset circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

19.1 Structure

The voltage detection circuit consists of a reference voltage generation circuit, comparators and control registers.

Supply voltage is divided by a ladder resistor and input to the voltage selection circuit. In the voltage selection circuit, a voltage is chosen according to the detected voltage then compared with the reference voltage in the comparator. If the supply voltage is upper/lower than the detected voltage, an interrupt signal occurs.

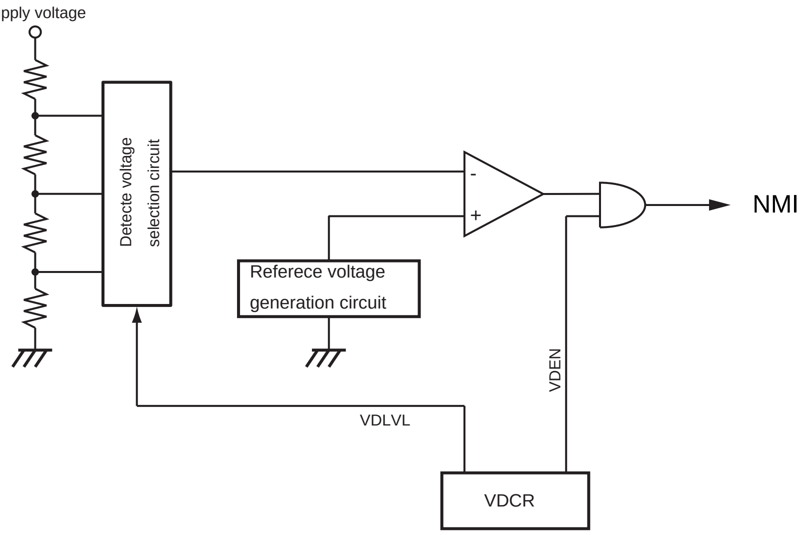

Supply voltage
Detecte voltage selection circuit
-
+
NMI
Referece voltage generation circuit
VDEN
VDLVL
VDCR

Figure 19-1 Block diagram of VLTD

19.2 Registers

19.2.1 Register List

Base Address = 0x4004_9000

Register name		Address (Base+)
Control register	VDCR	0x0000
Status register	VDSR	0x0004

19.2.2 VDCR (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	0	0

Bit	Bit Symbol	Type	Function
31 - 3	−	R	Read as "0".
2 - 1	VDLVL[1:0]	R/W	Detection voltage 00: Reserved 01: 4.1 ± 0.2V 10: 4.4 ± 0.2V 11: 4.6 ± 0.2V
0	VDEN	R/W	Voltage detection operation 0: Disabled 1: Enabled

Note: VDCR is initialized by power-on reset and reset with reset pin.

19.2.3 VDSR (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	−	−	−	−	−	−	−	VDSR
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31 - 1	−	R	Read as "0".
0	VDSR	R	Voltage detection status 0: Power supply voltage is upper than the detection voltage. 1: Power supply voltage is lower than the detection voltage.

19.3 Operation Description

19.3.1 Detection Voltage Selection and Enabling/Disabling the Operation

The VDCR register sets the following; choosing the detection voltage, setting the operation to enable/disable, choosing output conditions and setting the output to enable/disable. The VDCR register is initialized with the power-on reset, the VLTD reset or the $\overline{\text{RESET}}$ by reset pin.

The VDCR<VDLVL[1:0]> bit chooses the detection voltage. If VDCR<VDEN> is set to "1", voltage detection operation is enabled.

Note: While supply voltage is lower than the detection voltage, if voltage detection operation is enabled, NMI will generate at this timing.

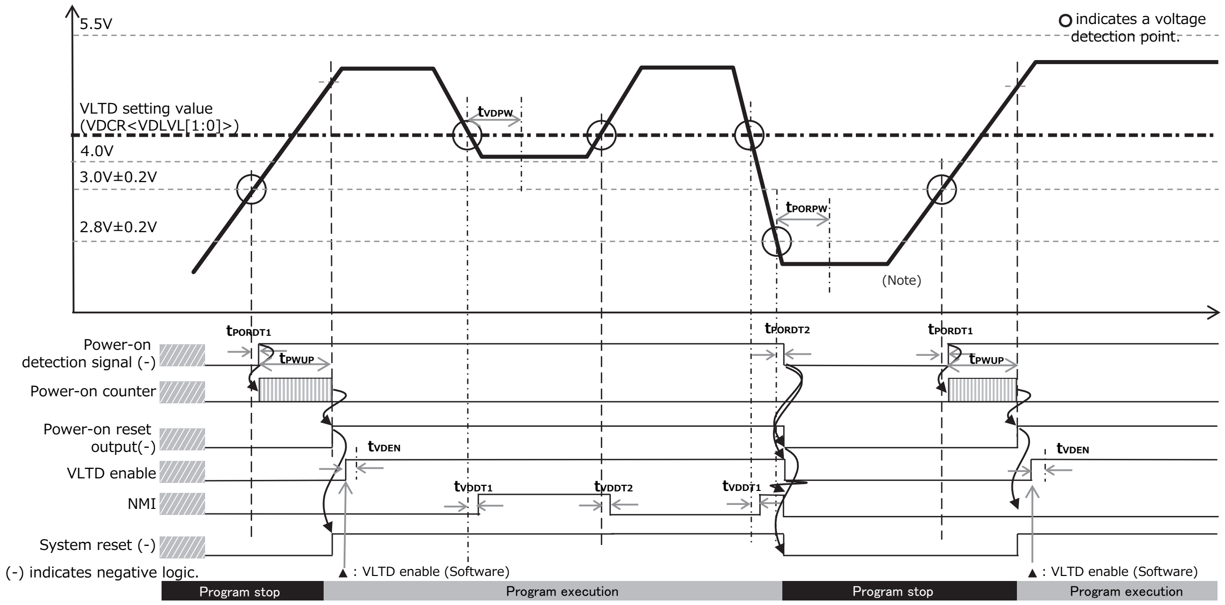
19.3.2 Lower Voltage Detection

If supply voltage is lower than the detection voltage level, NMI occurs. As long as supply voltage is lower than the detection voltage level, NMI is generated continuously.

It is required to give a certain time until NMI is detected when a low voltage is detected. If this period is shorter than expected, NMI may not occur.

Once the supply voltage is lower than 4.0V, the detection operation is not guaranteed. In this case, decrease the supply voltage to 0V and then power it on.

19.3.3 Operation Example of Power-on Reset and VLTD



Note: Once the supply voltage is lower than 4.0V, the detection operation is not guaranteed. In this case, decrease the supply voltage to 0V and then power it on.

Figure 19-2 Operation Example of Power-on Reset and VLTD

19.3.3.1 POR

Parameter	Symbol	Min	Typ.	Max	Unit
POR release response time	t_{PORDT1}	–	30	–	us
POR detection response time	t_{PORDT2}	–	30	–	
POR detection pulse width	t_{PORPW}	45	–	–	
Power on counting time	t_{PWUP}	–	$2^{13}/f_{OSC2}$ (Note)	–	ms

Note:f_{OSC2}=9.0MHz ± 15%

19.3.3.2 VLTD

Parameter	Symbol	Min	Typ.	Max	Unit
VLTD detection response time	t_{VDDT1}	–	40	–	us
VLTD release response time	t_{VDDT2}	–	40	–	
VLTD detection pulse width	t_{VDPW}	45	–	–	
A time to become the circuit operation valid after the LVTD is enabled.	t_{VEDN}	–	40	–	

20. Oscillation Frequency Detector (OFD)

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

The OFD monitors the target clock frequency using reference frequency and generates a reset signal if abnormal state is detected. And if the reference frequency stops, a reset signal is generated, too.

TMPM3U6FY/FW uses internal high-speed oscillator clock as a reference and the target clock is an external high-speed oscillator clock.

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

20.1 Block diagram

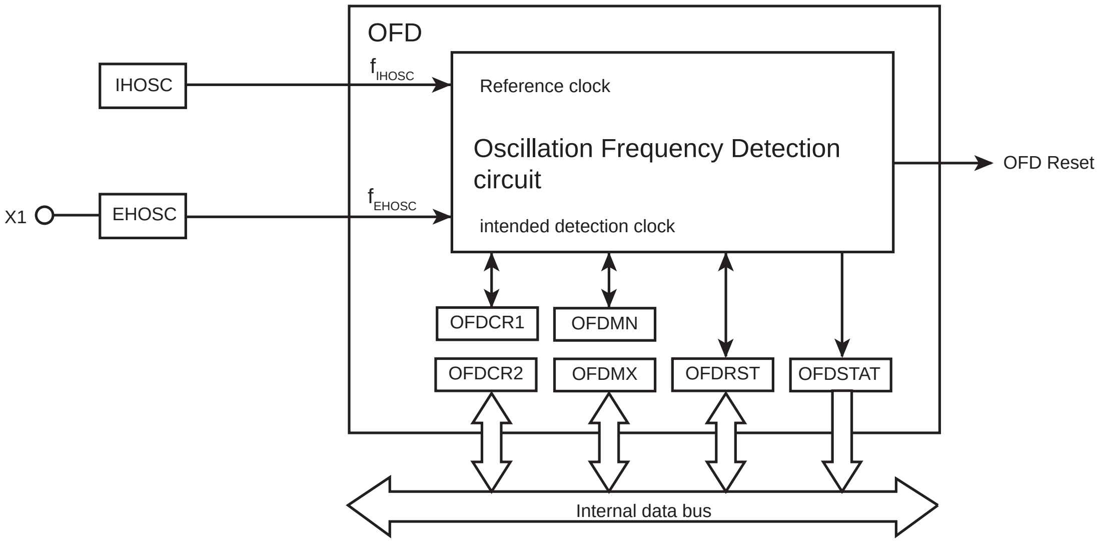


Figure 20-1 Oscillation Frequency Detector Block diagram

20.2 Registers

20.2.1 Register List

Base Address = 0x4004_0800

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

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

20.2.1.1 OFDCR1 (Control register 1)

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

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

20.2.1.2 OFDCR2 (Control register 2)

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

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

20.2.1.3 OFDMN (Lower detection frequency setting register)

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

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDMN[7:0]	R/W	Sets lower detection frequency.

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

20.2.1.4 OFDMX (Higher detection frequency setting register)

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

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDMX[7:0]	R/W	Sets higher detection frequency.

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

20.2.1.5 OFDRST (Reset control register)

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

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

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

20.2.1.6 OFDSTAT (Status register)

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

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

20.3 Operational Description

20.3.1 Setting

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

The frequency range to be detected is set with OFDMX and OFDMN. Set OFDRST enable reset generation and write "0xE4" to OFDCR2 to start operation.

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

20.3.2 Operation

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

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

When generating reset is enabled, the reset is generated if the following condition is satisfied.

- When a target clock frequency is over than the range of a frequency which is specified by OFDMX and OFDMN.
- When the reference clock stops.

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

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

20.3.3 Detection Frequency

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

Figure 20-2 shows the detection or undetectable frequency range when the target clock error is ±10% and the reference clock error is ± 5%.

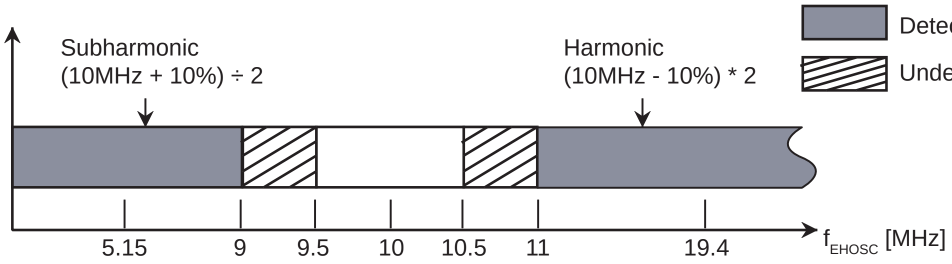


Figure 20-2 Example of detection frequency range (in case of 10MHz)

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

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

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

higher limit of the detection frequency	$= 1 \div \{(d \div 2^8) \div (a \div 4)\}$	
lower limit of the detection frequency	$= 1 \div \{(c \div 2^8) \div (b \div 4)\}$	
higher limit of the detection frequency	$= 1 \div \{(9.5 \times 10^6 \div 2^8) \div (11 \times 10^6 \div 4)\}$	= 74.10 = 74 (truncate after the decimal places)= 0x4A
lower limit of the detection frequency	$= 1 \div \{(10.5 \times 10^6 \div 2^8) \div (9 \times 10^6 \div 4)\}$	= 54.85 = 55 (round up after the decimal places)= 0x37

Setting "0x4A" to the register OFDMX and "0x37" to the register OFDMN, when the external oscillation of higher than 11MHz or lower than 9MHz is detected, the oscillation frequency detector outputs a reset signal.

20.3.4 Available Operation Mode

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

20.3.5 Example of Operational Procedure

The example of operational procedure is shown below.

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

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

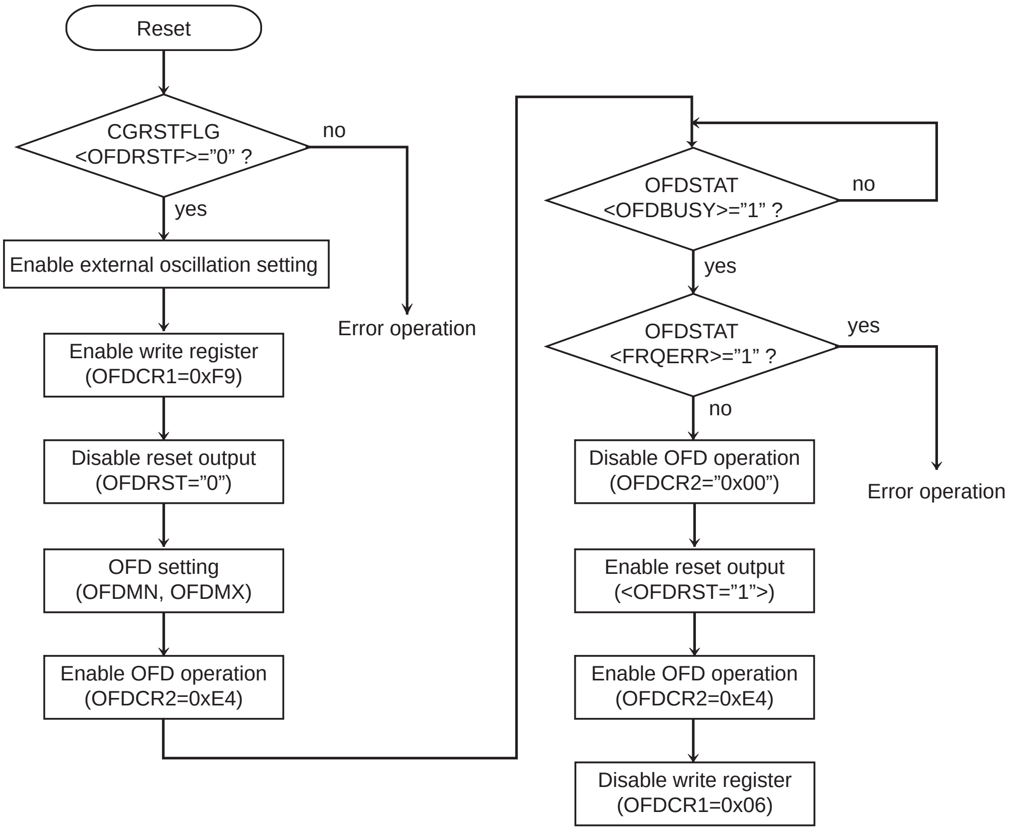

Reset
CGRSTFLG <OFDRSTF>="0" ?
no
yes
Enable external oscillation setting
Error operation
Enable write register (OFDCR1=0xF9)
Disable reset output (OFDRST="0")
OFD setting (OFDMN, OFDMX)
Enable OFD operation (OFDCR2=0xE4)
OFDSTAT <OFDBUSY>="1" ?
no
yes
OFDSTAT <FRQERR>="1" ?
yes
no
Disable OFD operation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

Figure 20-3 Example of operational procedure

21. Watchdog Timer(WDT)

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

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

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

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

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

21.1 Configuration

Figure 21-1shows the block diagram of the watchdog timer.

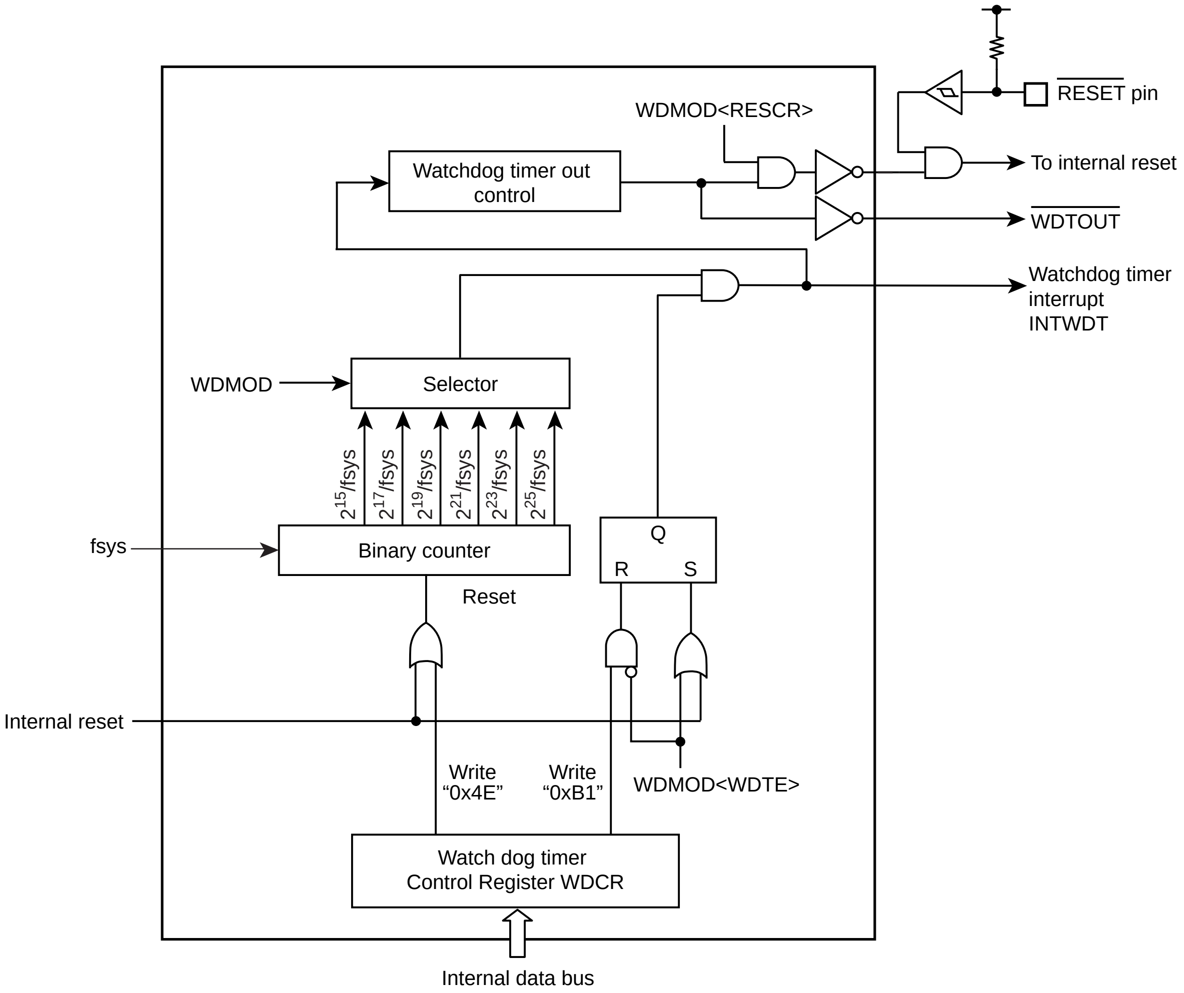


Figure 21-1 Block Diagram of the Watchdog Timer

21.2 Register

The followings are the watchdog timer control registers and addresses.

Base Address = 0x4004_0000

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

21.2.1 WDMOD(Watchdog Timer Mode Register)

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

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

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

Table 21-1 Detection time of watchdog timer (fc = 40 MHz)

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

21.2.2 WDCR (Watchdog Timer Control Register)

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

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

21.3 Operations

21.3.1 Basic Operation

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

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

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

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

21.3.2 Operation Mode and Status

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

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

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

- STOP mode
- SLEEP mode
- SLOW mode

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

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

21.4 Operation when malfunction (runaway) is detected

21.4.1 INTWDT interrupt generation

In the Figure 21-2 shows the case that INTWDT interrupt generates (WDMOD<RESCR>="0").

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

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

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

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

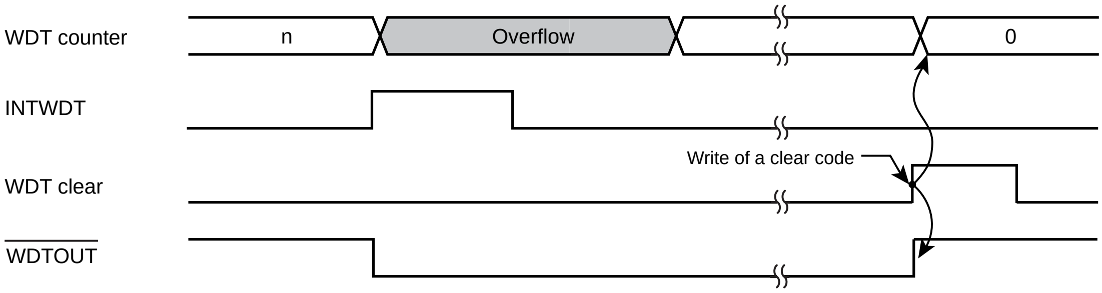


Figure 21-2 INTWDT interrupt generation

21.4.2 Internal reset generation

Figure 21-3 shows the internal reset generation (WDMOD<RESCR>="1").

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

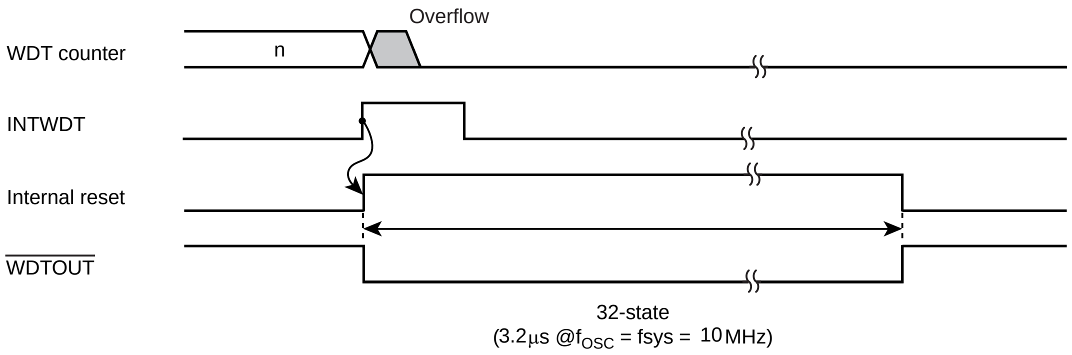

Overflow
WDT counter
n
INTWDT
Internal reset
$\overline{\text{WDTOUT}}$
32-state
(3.2μs @f_{OSC} = fsys = 10 MHz)

Figure 21-3 Internal reset generation

21.5 Control register

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

21.5.1 Watchdog Timer Mode Register (WDMOD)

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

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

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

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

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

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

3. Watchdog timer out reset connection <RESCR>

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

21.5.2 Watchdog Timer Control Register(WDCR)

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

21.5.3 Setting example

21.5.3.1 Disabling control

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

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

21.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

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

21.5.3.3 Watchdog timer clearing control

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

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

21.5.3.4 Detection time of watchdog timer

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

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

22. Flash

This section describes the hardware configuration and operation of the flash memory.

22.1 Flash Memory

22.1.1 Features

1. Memory capacity

   TMPM3U6FY/FW contains flash memory. The memory sizes and configurations are shown in the table below.

   Independent write access to each block is available. When the CPU is to access the internal flash memory, 32-bit data bus width is used.

2. Write / erase time

   Writing is executed per page. TMPM3U6FY/FW contains 64 words.

   Page writing requires 1.25ms (typical) regardless of number of words. A

   block erase requires 0.1 s. (typical).

   The following table shows write and erase time per chip.

Memory size	Block Configuration				# of words	Write time	Erase time
	128 KB	64 KB	32 KB	16 KB			
256 KB	0	3	1	2	64	1.28 s	0.4 s
128 KB	0	1	1	2	64	0.64 s	0.4 s

Note: **The above value are theoretical value not including data transfer time. The write time per chip depends on the write method to be used by users.**

3. Programming method

   There are two types of the onboard programming mode for users to program (rewrite) the device while it is mounted on the user's board:

   a. User boot mode

      The use’s original rewriting method can be supported.

   b. Single boot mode

      The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

4. Rewriting method

The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if a user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.

JEDEC compliant functions	Modified, added, or deleted functions
・ Automatic programming ・ Automatic chip erase ・ Automatic block erase ・ Data polling / toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

5. Protect/ Security Function

This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See the chapter "ROM protection" for details of ROM protection and security function.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

22.1.2 Block Diagram of the Flash Memory Section

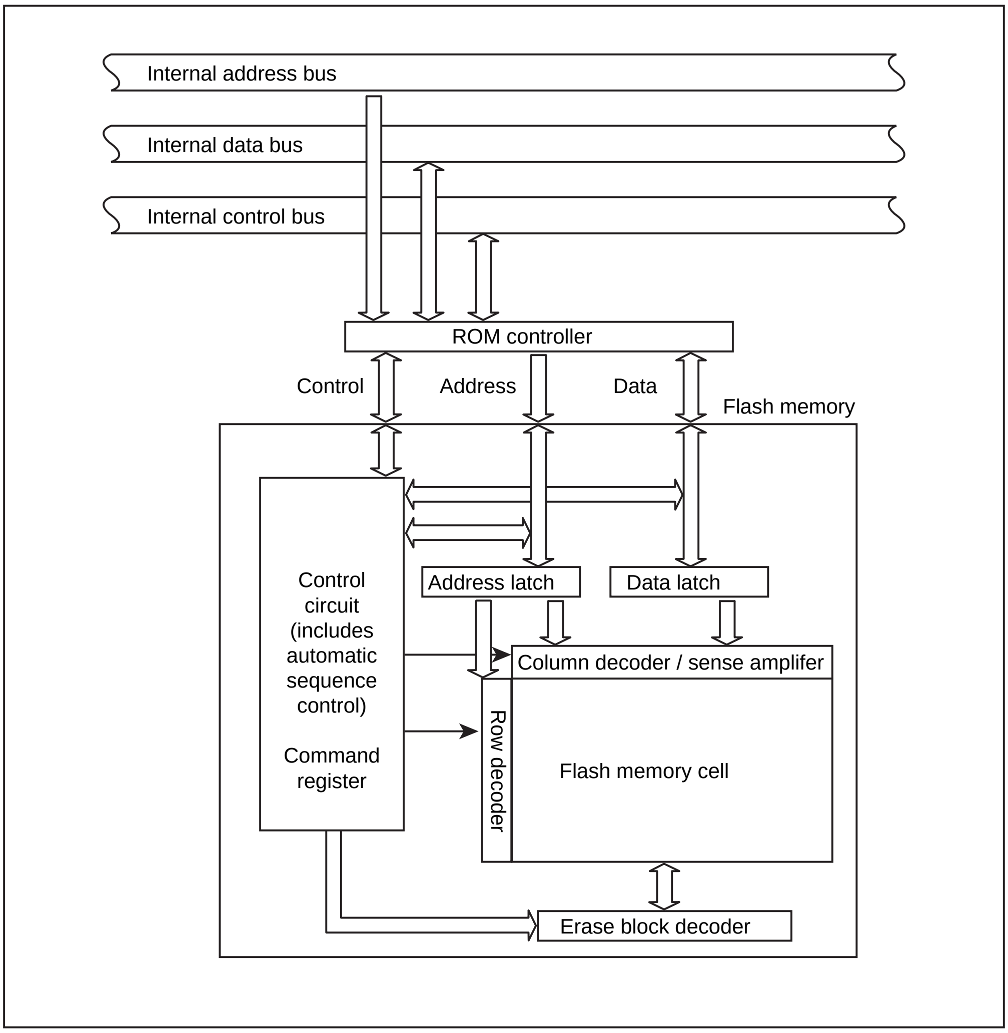


Figure 22-1 Block Diagram of the Flash Memory Section

22.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 22-1 Operation modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's set, are defined. The former is referred to as "normal mode" and the latter "user boot mode".
	User boot mode	A user can uniquely configure the system to switch between these two modes. For example, a user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0". A user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the Table 22-1, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's set.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ pin while the device is in reset status.

Table 22-2 Operating Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

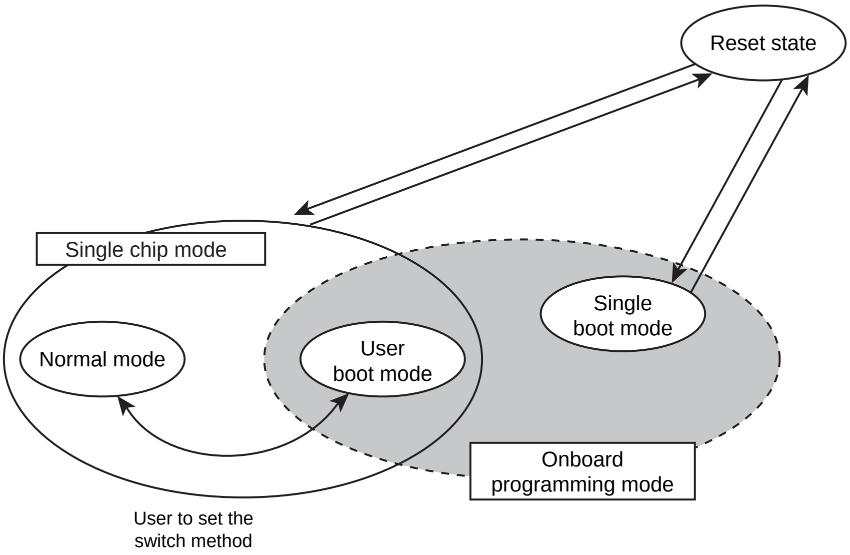


Figure 22-2 Mode Transition Diagram

22.2.1 Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the RESET input is held at "0" for a minimum duration of 12 system clocks (0.3µs with 40MHz operation; the "1/1" clock gear mode is applied after reset).

Note 1: **It is necessary to apply "0" to the RESET inputs upon power on for a minimum duration of 300 µs regardless of the operating frequency.**

Note 2: **While flash auto programming or erasing is in progress, at least 0.5 µs of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

22.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the user application.

The condition to switch the modes needs to be set by using the I/O of TMPM3U6FY/FW in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete / writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. Be sure not to cause any exceptions including a non-maskable while User Boot Mode.

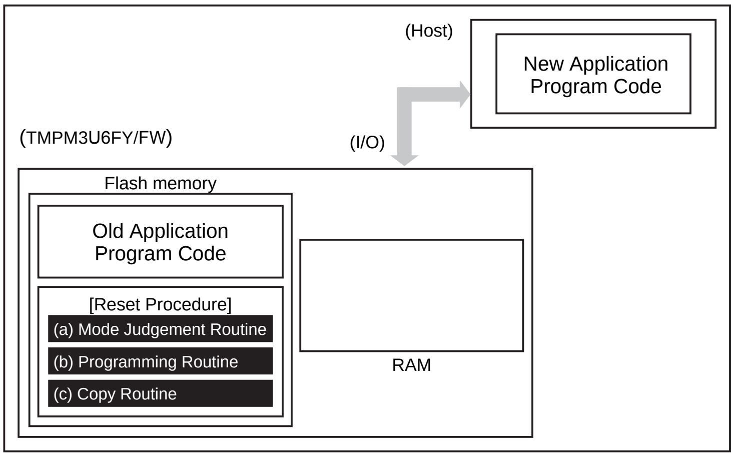
(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "22.3 On-board Programming of Flash Memory (Rewrite/Erase)".

22.2.2.1 (1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Step-1

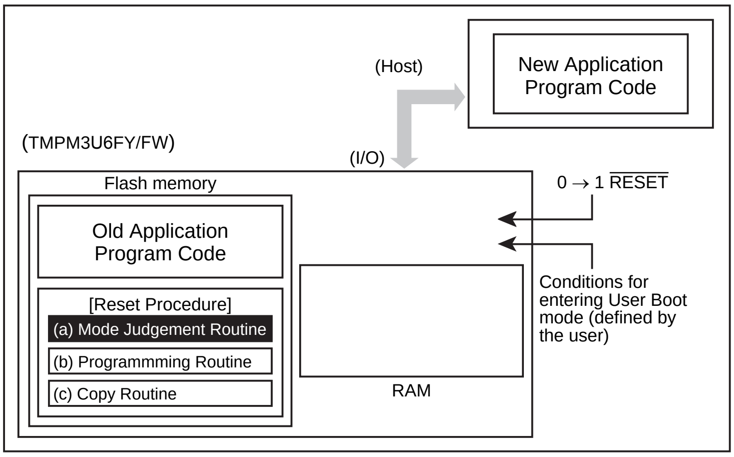
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM3U6FY/FW on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine:	Code to determine whether or not to switch to User Boot mode
(b) Programming routine:	Code to download new program code from a host controller and re-program the flash memory
(c) Copy routine:	Code to copy the data described in (b) from the TMPM3U6FY/FW flash memory to either the TMPM3U6FY/FW on-chip RAM or external memory device.



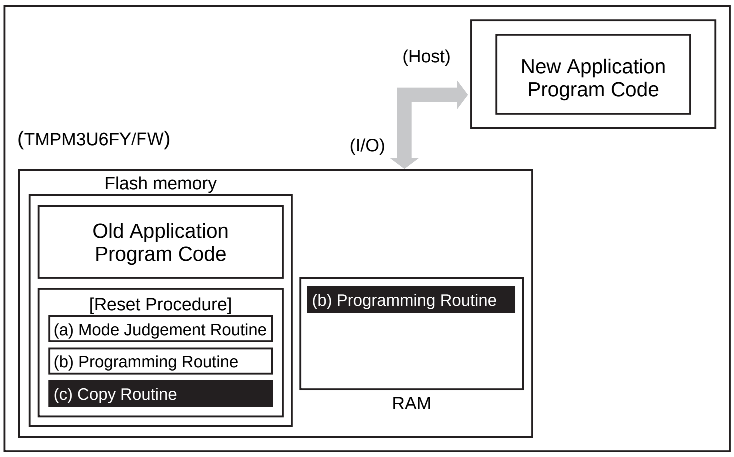
(2) Step-2

The following description is the case that programming routines are installed in the reset processing program. After RESET pin is released, the reset procedure determines whether to put the TMPM3U6FY/FW flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be not used while in User Boot mode.)


(Host)
New Application Program Code
(TMPM3U6FY/FW)
(I/O)
Flash memory
0 → 1 RESET
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Programmming Routine
(c) Copy Routine
RAM
Conditions for entering User Boot mode (defined by the user)

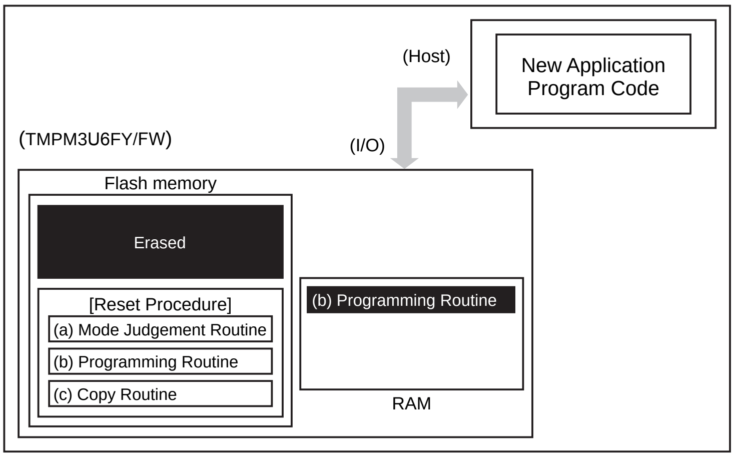
(3) Step-3

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM3U6FY/FW on-chip RAM.



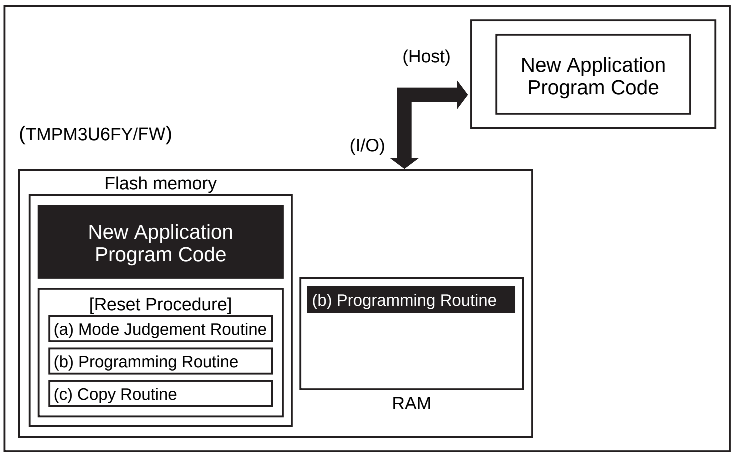
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to clear write or erase protection and erase a flash block containing the old application program code.



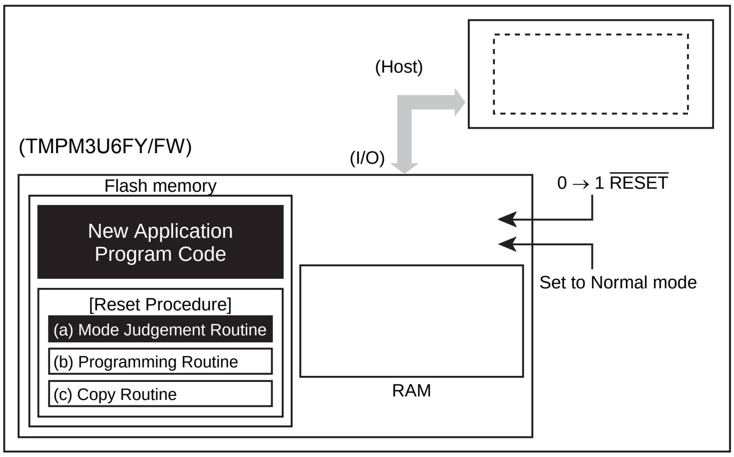
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user’s program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM3U6FY/FW. Upon reset, the on-chip flash memory is set to Normal mode. After RESET is released, the CPU will start executing the new application pro-gram code.



22.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

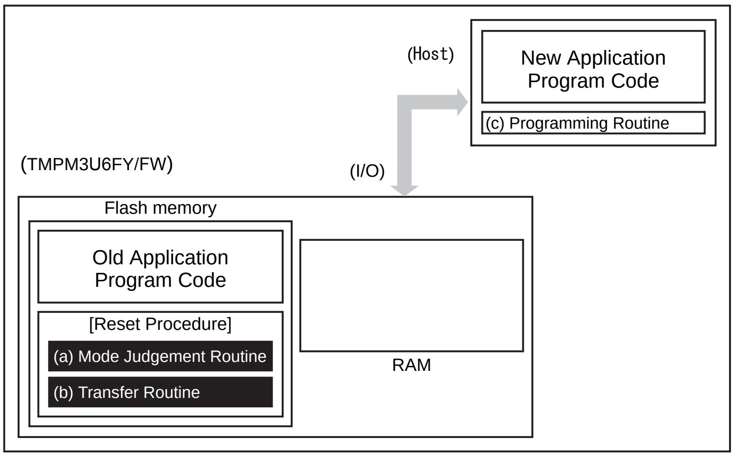
(1) Step-1

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM3U6FY/FW on a printed circuit board, write the following pro-gram routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine:	Code to determine whether or not to switch to User Boot mode
(b) Transfer routine:	Code to download new program code from a host controller

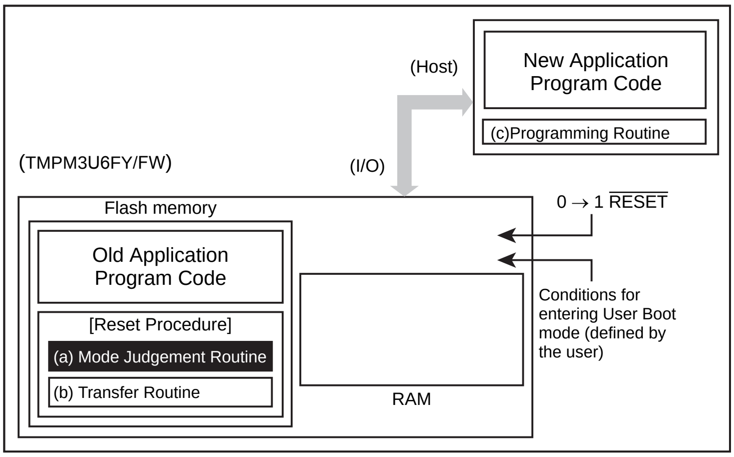
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine:	Code to download new program code from an external host controller and re-program the flash memory



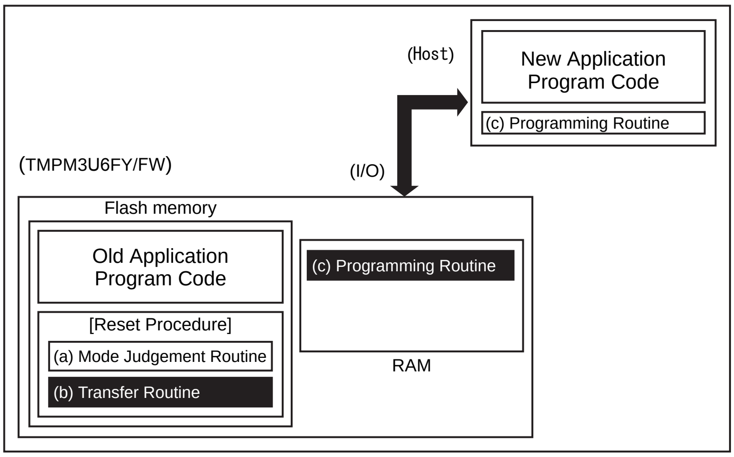
(2) Step-2

The following description is the case that programming routines are installed in the reset processing program. After RESET is released, the reset procedure determines whether to put the TMPM3U6FY/FW flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be not used while in User Boot mode).


(Host)
New Application Program Code
(c)Programming Routine
(TMPM3U6FY/FW)
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old Application Program Code
[Reset Procedure]
(a) Mode Judgement Routine
(b) Transfer Routine
RAM
Conditions for entering User Boot mode (defined by the user)

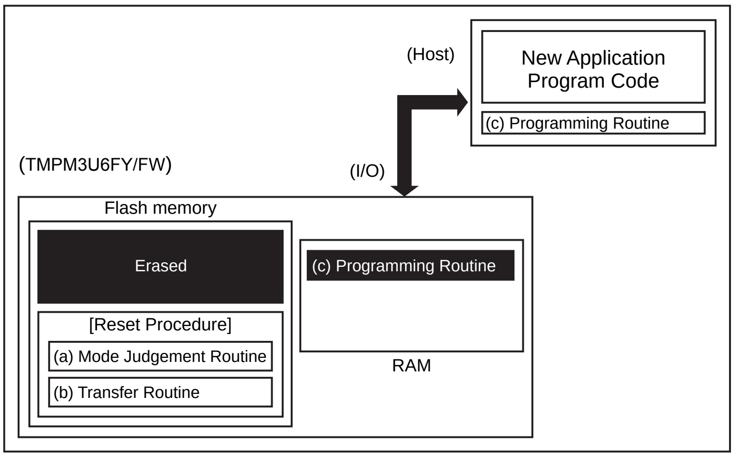
(3) Step-3

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM3U6FY/FW on-chip RAM.



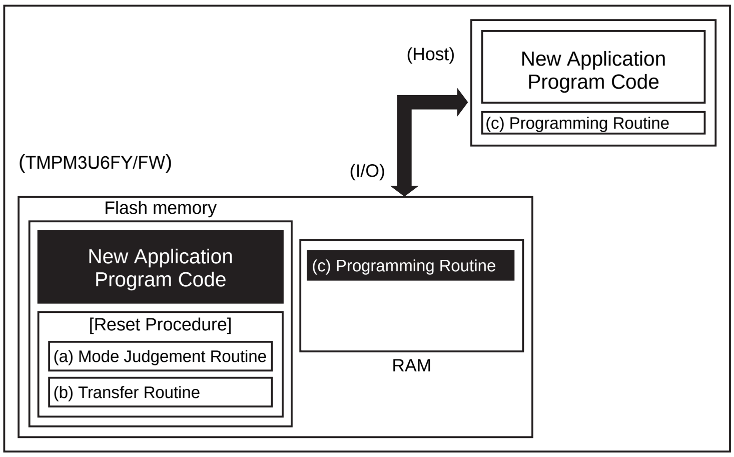
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to clear write or erase protection and erase a flash block containing the old application program code.



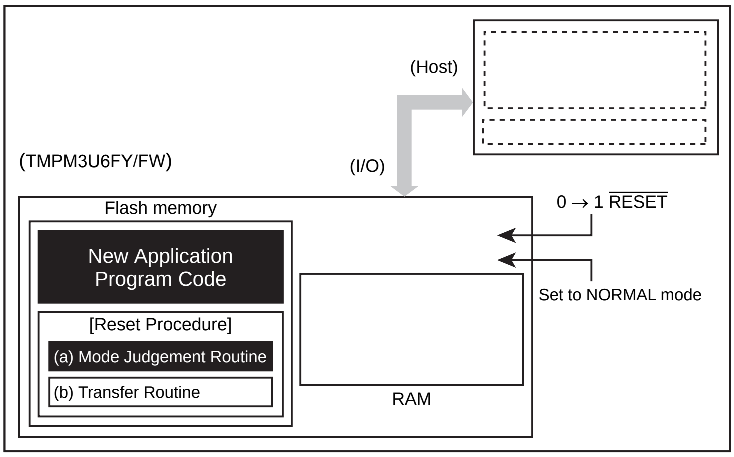
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM3U6FY/FW. Upon reset, the on-chip flash memory is set to Normal mode. After RESET is released, the CPU will start executing the new application program code.



22.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM3U6FY/FW on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it.

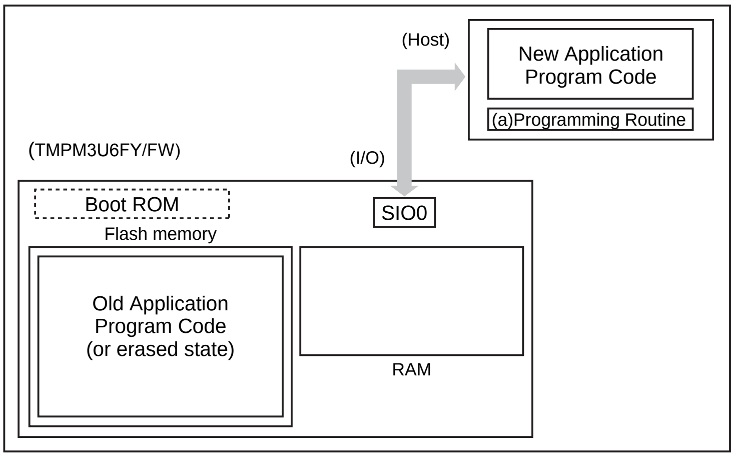
Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMPM3U6FY/FW are connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM3U6FY/FW on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and program-ming data to re-program the flash memory. Communications between the SIO0 and the host must follow the pro-tocol described later. To secure the contents of the flash memory, the validity of the application's password is verified before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted. As in the case of User Boot mode, all interrupts includ-ing the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is be-ing erased or programmed. In Single Boot mode, the boot-ROM programs are executed in Normal mode.

Once re-programming is complete, it is recommended to set the write/erase protection to the relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

22.2.3.1 (2-A) Using the Program in the On-Chip Boot ROM

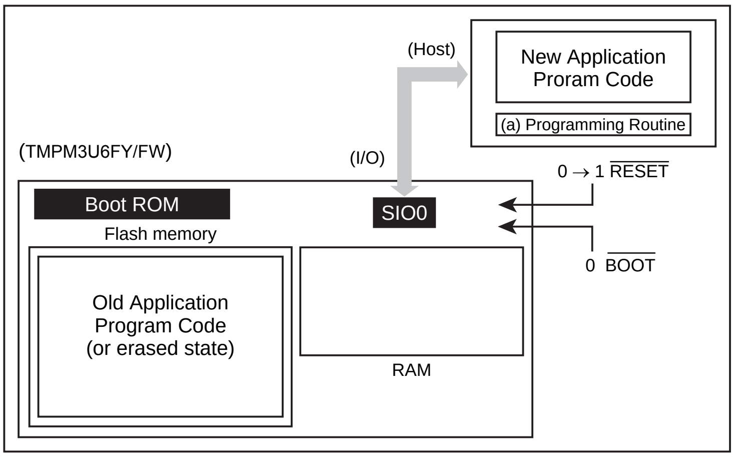
(1) Step-1

The flash block containing the old version of the program code does not need to be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.


(Host)
New Application Program Code
(a)Programming Routine
(TMPM3U6FY/FW)
(I/O)
Boot ROM
Flash memory
SIO0
Old Application Program Code (or erased state)
RAM

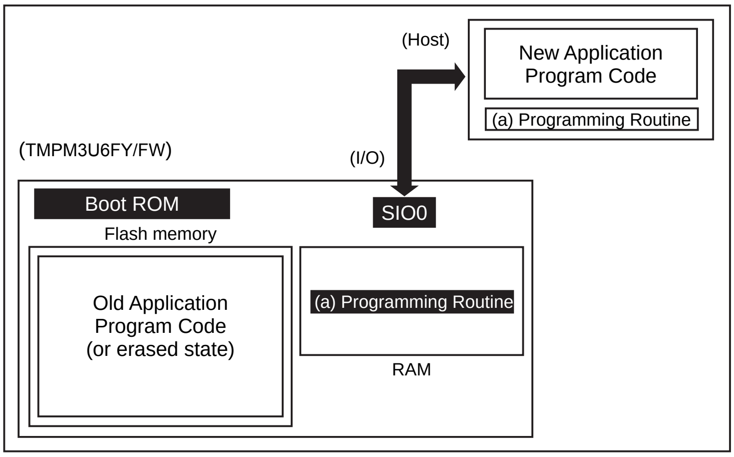
(2) Step-2

Set the $\overline{\text{RESET}}$ pin to "1" to cancel the reset of the TMPM3U6FY/FW when the $\overline{\text{BOOT}}$ pin has already been set to "0". After reset, CPU reboots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is firstly compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFF).



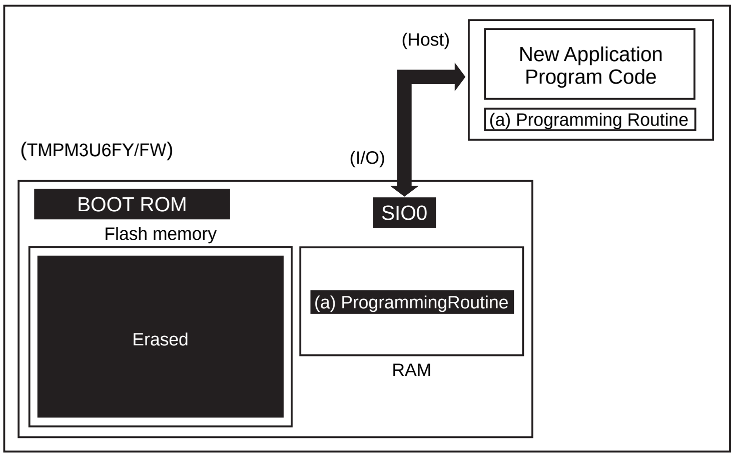
(3) Step-3

If the password is correct, the boot program downloads the programming routine (a) from the host controller into the on-chip RAM of the TMPM3U6FY/FW. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



(4) Step-4

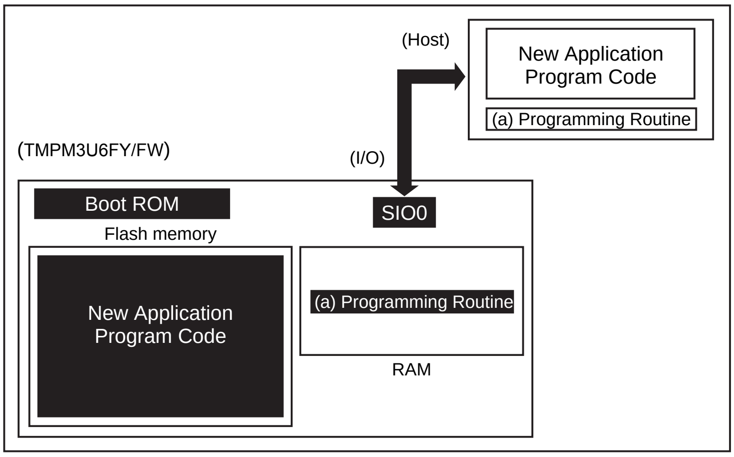
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



(5) Step-5

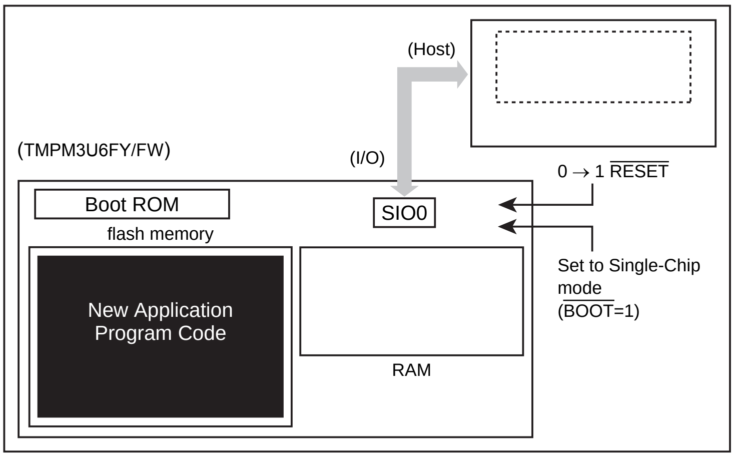
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute in the on-chip RAM, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) Step-6

When programming of the flash memory is complete, power off the board and disconnect the cable between the host and the target board. Turn on the power again so that the TMPM3U6FY/FW re-boots in Single-Chip (Normal) mode to execute the new program.


(Host)
(TMPM3U6FY/FW)
(I/O)
0 → 1 $\overline{\text{RESET}}$
Boot ROM
SIO0
flash memory
New Application Program Code
RAM
Set to Single-Chip mode ($\overline{\text{BOOT}}$=1)

22.2.4 Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM3U6FY/FW with Single Boot mode following the configuration shown below.

$\overline{\text{BOOT}}$ = 0
$\overline{\text{RESET}}$ = 0 → 1

Set the $\overline{\text{RESET}}$ input pin to "0", and set the each $\overline{\text{BOOT}}$ pins to values shown above, and then release $\overline{\text{RESET}}$ pin (high).

22.2.5 Memory Map

Figure 22-3 and Figure 22-4 show a comparison of the memory maps in Single chip and Single Boot modes. In Single Boot mode, the internal flash memory is mapped to 0x3F80_0000 and later addresses, and the Internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_0FFF.

The internal flash memory and RAM addresses of each device are shown below.

Flash Size	RAM Size	Flash Address (Single Chip / Single Boot Mode)	RAM Address
256 KB	16 KB	0x0000_0000 to 0x0003_FFFF / 0x3F80_0000 to 0x3F83_FFFF	0x2000_0000 to 0x20000_3FFF
128K	12 KB	0x0000_0000 to 0x0001_FFFF / 0x3F80_0000 to 0x3F81_FFFF	0x2000_0000 to 0x2000_2FFF

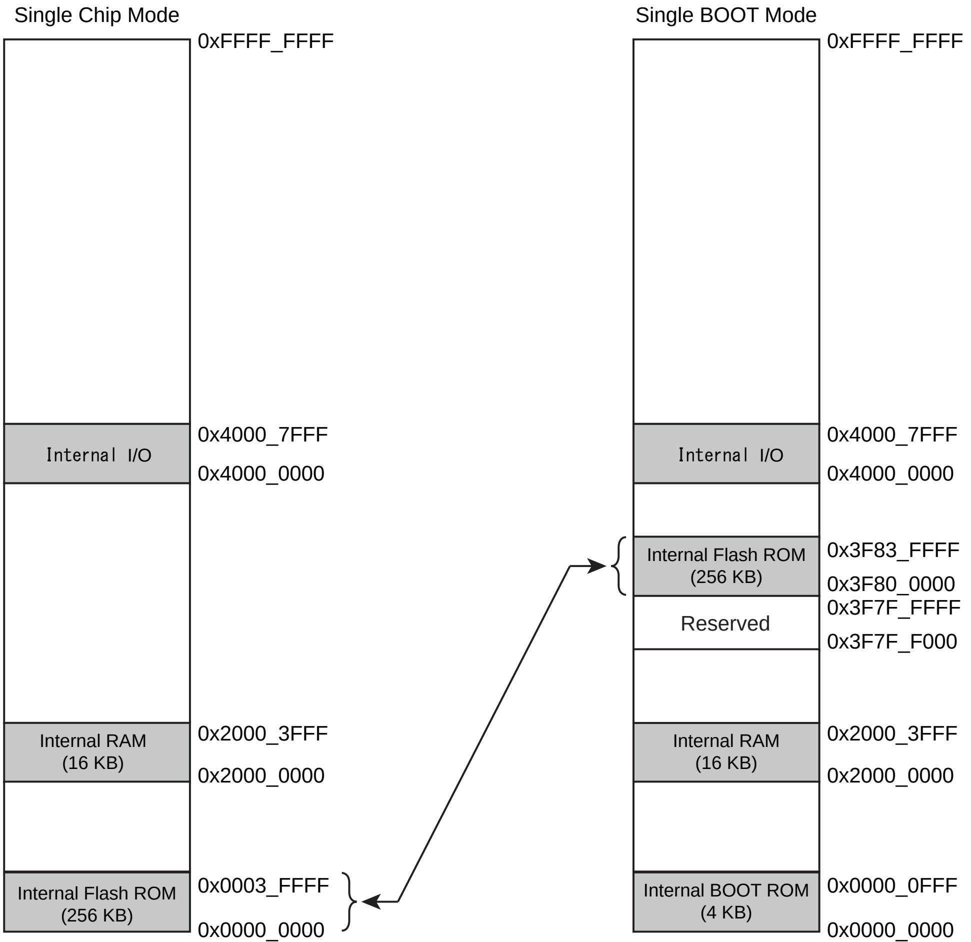


Figure 22-3 Memory Maps for TMPM3U6FY

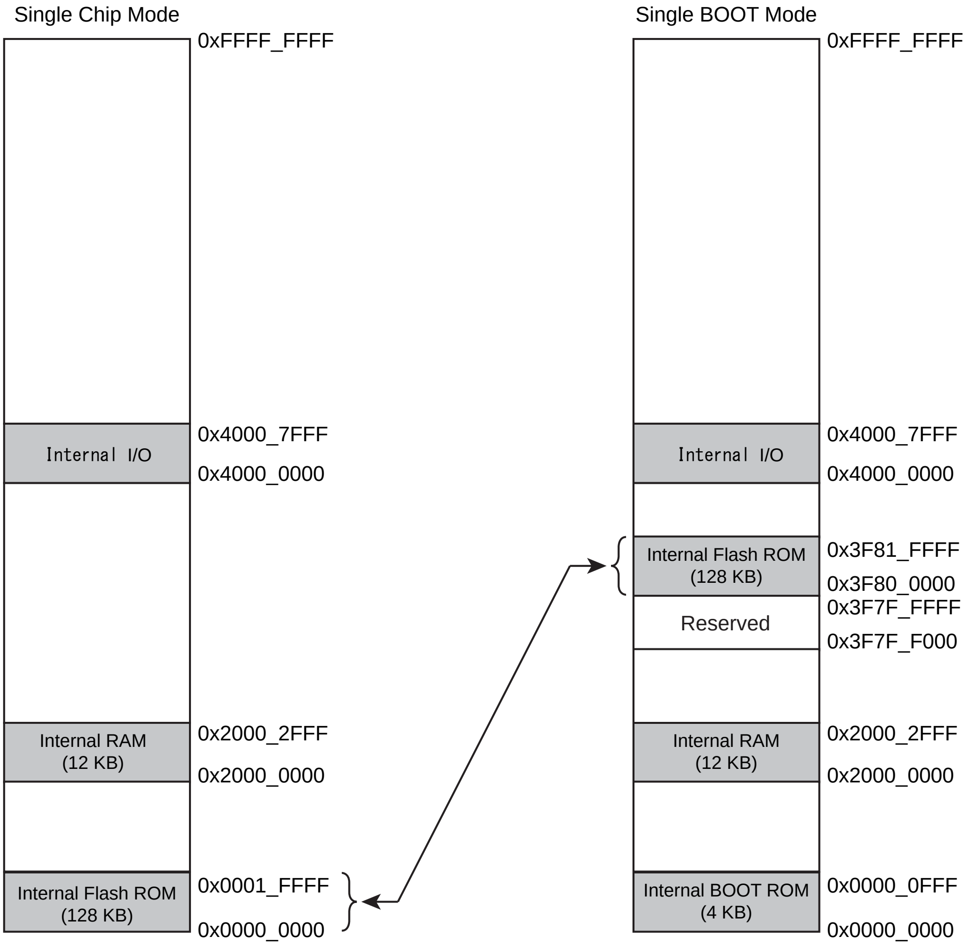


Figure 22-4 Memory Maps for TMPM3U6FW

22.2.6 Interface specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. The same configuration is applied to a communication format on a programming controller to execute the on-board programming. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below.

- UART communication

  Communication channel : SIO channel 0

  Serial transfer mode : UART (asynchronous), half -duplex, LSB first

  Data length : 8 bits

  Parity bit : None

  STOP bit : 1 bit

  Baud rate : Arbitrary baud rate

- I/O Interface mode

  Communication channel : SIO channel 0

  Serial transfer mode : I/O interface mode, full -duplex, LSB first

  Synchronization clock (SCLK0) : Input mode

  Handshaking signal : PE4 configured as an output mode

  Baud rate : Arbitrary baud rate

Table 22-3 Required Pin Connections

Pin		Interface	
		UART	I/O Interface Mode
Power supply pins	DVDD5	o	o
	REGVDD5	o	o
	AVDD	o	o
	VOUT3	o	o
	DVSS	o	o
	AVSS	o	o
	CVSS	o	o
Mode-setting pin	$\overline{\text{BOOT}}$	o	o
Reset pin	$\overline{\text{RESET}}$	o	o
Communication pin	TXD0 (PE0)	o	o
	RXD0 (PE1)	o	o
	SCLK0 (PE2)	×	o (Input mode)
	PE4	×	o (Output mode)

22.2.7 Data Transfer Format

Table 22-4, Table 22-6 to Table 22-7 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to "22.2.10 Operation of Boot Program".

Table 22-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x40	Chip and protection bit erase

22.2.8 Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 22-5.

Table 22-5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	A program contained in the BOOT ROM uses the area, through 0x2000_0000 to 0x2000_03FF, as a work area. Store the RAM transfer program from 0x2000_0400 through the end address of RAM.
Internal ROM	The following addresses are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable. 0x3F83_FFF0 to 0x3F83_FFFF

22.2.9 Transfer Format for Boot Program

The following tables shows the transfer format for each Boot program command. Use this section in conjunction with Chapter "22.2.10 Operation of Boot Program".

22.2.9.1 RAM Transfer

Table 22-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM3U6FY/FW	Baud rate	Data Transferred from the TMPM3U6FY/FW to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	−
	2 byte	−		ACK for the serial operation mode byte ・For UART mode - Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode - Normal acknowledge :0x30
	3 byte	Command code (0x10)		−
	4 byte	−		ACK for the command code byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	5 byte to 16 byte	Password sequence (12 bytes)) 0x3F87_FFF4 to 0x3F87_FFFF		−
	17 byte	Check SUM value for bytes 5 to 16		−
	18 byte	−		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	19 byte	RAM storage start address 31 to 24		−
	20 byte	RAM storage start address 23 to 16		−
	21 byte	RAM storage start address 15 to 8		−
	22 byte	RAM storage start address 7 to 0		−
	23 byte	RAM storage start address 15 to 8		−
	24 byte	RAM storage start address 7 to 0		−
	25 byte	Check SUM value for bytes 19 to 24		−
	26 byte	−		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	27 byte to mbyte	RAM storage data		−
	m+ 1 byte	Checksum value for bytes 27 to m		−
	m+ 2 byte	−		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
RAM	m+ 3 byte	−		Jump to RAM storage start address

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

Note 3: **The 19th to 25th bytes must be within the RAM address range from 0x2000_0400 through the end address of RAM.**

22.2.9.2 Chip Erase and Protect Bit Erase

Table 22-7 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM3U6FY/FW	Baud rate	Data Transferred from the TMPM3U6FY/FW to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	–
	2 byte	–		ACK for the serial operation mode byte ・For UART mode - Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) ・For I/O Interface mode - Normal acknowledge :0x30
	3 byte	Command code (0x40)		–
	4 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x40 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	5 byte	Chip erase command code (0x54)		–
	6 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x54 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	7 byte	–		ACK for the chip erase command code byte - Normal acknowledge : 0x4F - Negative acknowledge : 0x4C
	8 byte	(Wait for the next command code.)		–

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second byte must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

22.2.10 Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these four commands, of which the details are provided on the following subsections.

1. RAM Transfer command

   The RAM Transfer command stores program code transferred from the host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the range from 0x2000_0400 to the end address of RAM, whereas the boot program area (0x2000_0000 to 0x2000_03FF) is unavailable. The user program starts at the assigned RAM address.

   The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 22.3. Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

2. Flash Memory Chip Erase and Protection Bit Erase command

   This command erases the entire area of the flash memory automatically. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. This command serves to recover boot programming operation when a user forgets the password. Therefore password verification is not executed.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

22.2.10.1 RAM Transfer Command

See Table 22-6 for the transfer format of this command.

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see "22.2.10.4 Determination of a Serial Operation Mode" described later. If the mode is determined as UART mode, the boot program checks if the baud rate setting can be performed. During the first-byte processing, receiving operation is prohibited. (SC0MOD0<RXE> = "0")
   - To communicate in UART mode

     The 1st byte is set to "0x86" and is transmitted from the controller to the target board at the specified baud rate by setting UART. If the serial operation mode is determined as UART, then the boot program checks if the baud rate setting can be performed. If that baud rate cannot be set, the boot program aborts and any subsequent communications cannot be done. Please refer to "Baud rate setting" for the method of judging whether the setting of the baud rate is possible.

   - To communicate in I/O Interface mode

     The 1st byte is set to "0x30" and is transmitted from the controller to the target board at 1/16 of the desired baud rate by the synchronous setting. Same as the 1st byte, a 1/16 of the specified baud rate is used in the 2nd transmission. From the 3rd byte (operation command data), users can transmit data at specified baud rate.

     In I/O interface mode, CPU considers the reception terminal to be an input port and monitors the level of I/O port. If the baud rate is high or operation frequency is high, CPU may not distinguish the level of I/O port. To avoid this situation, the baud rate is set at the 1/16 of desired baud rate in the I/O interface. When the serial operation mode is determined as I/O Interface mode, SCLK Input mode is set. The controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no error acknowledge response (bit 3, 0x08).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte where the serial operation mode is set. When 1st byte is determined as UART and can be set at the specified baud rate, data "0x86" is transmitted. When 1st byte is determined as I/O interface, data "0x30" is transmitted.

   - UART mode

     The 2nd byte is used for distinguishing whether the baud rate can be set. If the baud rate can be set, a value of SC0BRCR is renewed and data "0x86" is sent to the controller. If the baud rate cannot be set, transmit operation is stopped and no data is transmitted. After transmission of 1st byte completed, the controller allows for five seconds of time-out. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. Receiving operation is permitted by setting SC0MOD0<RXE>=1, before loading 0x86 to the SIO transmit buffer.

   - I/O Interface mode

     The boot program sets a value of the SC0MOD0 and SC0CR registers to configure the I/O Interface mode and writes 0x30 to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. After the transmission of the 1st byte completed, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 of the desired baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller regards it as communication possible. From the 3rd byte, users can transmit data at specified baud rate. Receiving operation is permitted by setting SC0MOD0<RXE>=1, before loading 0x86 to the SIO.

3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there is a receive error, the boot program transmits 0xX8 (bit 3) and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 22-4, the boot program echoes it back to the controller. When the RAM Transfer command is received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in the later Section "Password". If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5th byte. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Area
TMPM3U6FY/FW	0x3F83_FFF4 to 0x3F83_FFFF

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, ignore the carries and

calculate the 8-bit two's complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in details in the later Section "Checksum Calculation".

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th byte. If there is a receive error, the boot program sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

   Next, the RAM Transfer routine performs the checksum operation to ensure 17th byte data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   Finally, the password verification result is checked. If the following case is generated, the boot program transmits an acknowledge response (bit 0, 0x11) as a password error and waits for next operation command (3rd byte).

   - Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
   - Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

   When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31 to 24 of the address and the 22nd byte corresponds to bits 7 to 0 of the address. The start address of the stored RAM must be even address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15 to 8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7 to 0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, ignore the carries and calculate the 8-bit two's complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in detail in the later Section "22.2.10.6 Checksum Calculation".

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there is a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 24th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

    - The 19th to 25th bytes data must be within the range of 0x2000_0400 to the end address of RAM.

    When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to (m)th bytes from the controller are stored in the on-chip RAM of the TMPM3U6FY/FW. Storage begins at the address specified by the 19th to 22nd bytes and continues for the number of bytes specified by the 23rd to 24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to (m) th bytes together, ignore the carries and calculate the 8-bit two’s complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in detail in later Section "22.2.10.6 Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes. First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there is a receive error, the RAM Transfer routine sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been completed successfully, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes.

22.2.10.2 Chip and Protection Bit Erase Command

See Table 22-7 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. From the Controller to the TMPM3U6FY/FW

   The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x40.

3. From TMPM3U6FY/FW to the Controller

   The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 3rd byte is equal to any of the command codes listed in Table 22-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x40. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. From the controller to the TMPM3U6FY/FW

   The 5th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).

5. From TMPM3U6FY/FW to the Controller

   The 6th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 5th byte is equal to any of the command codes to enable erasing, the boot program echoes it back to the controller. When the Chip and Protection Erase command was received, the boot program echoes back a value of 0x54 and then branches to the Chip Erase routine. If the 5th byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

6. From TMPM3U6FY/FW to the Controller

   The 7th byte indicates whether the Chip Erase command is normally completed or not.

   At normal completion, completion code (0x4F) is sent.

   When an error was detected, error code (0x4C) is sent.

7. The 8th byte is the next command code.

22.2.10.3 Acknowledge Responses

The boot program represents processing states with specific codes. Table 22-8 to show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. The 3rd bit indicates a receive error. The 0th bit indicates an invalid command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not done in I/O Interface mode.

Table 22-8 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

Note: **In the UART mode, if the baud rate setting cannot be set, the communication is stopped without any response.**

Table 22-9 ACK Response to the Command Byte

Return Value	Meaning
0xN8 (See note)	A receive error occurred while receiving a command code.
0xN1 (See note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.
0x40	The Chip Erase command was received.

Note: **The upper four bits of the ACK response are the same as those of the previous command code.**

Table 22-10 ACK Response to the Checksum Byte

Return Value	Meaning
0xN8 (See note)	A receive error occurred.
0xN1 (See note)	A checksum or password error occurred.
0xN0 (See note)	The checksum was correct.

Note: **The upper four bits of the ACK response are the same as those of the operation command code. For example, it is 1 (N ; RAM transfer command data [7:4]) when password error occurs.**

Table 22-11 ACK Response to Chip and Protection Bit Erase Byte

Return Value	Meaning
0x54	The Chip Erase enabling command was received.
0x4F	The Chip Erase command was completed.
0x4C	The Chip Erase command was abnormally completed.

22.2.10.4 Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must firstly send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 of the desired baud rate. Figure 22-5 shows the waveforms for the first byte in each mode.

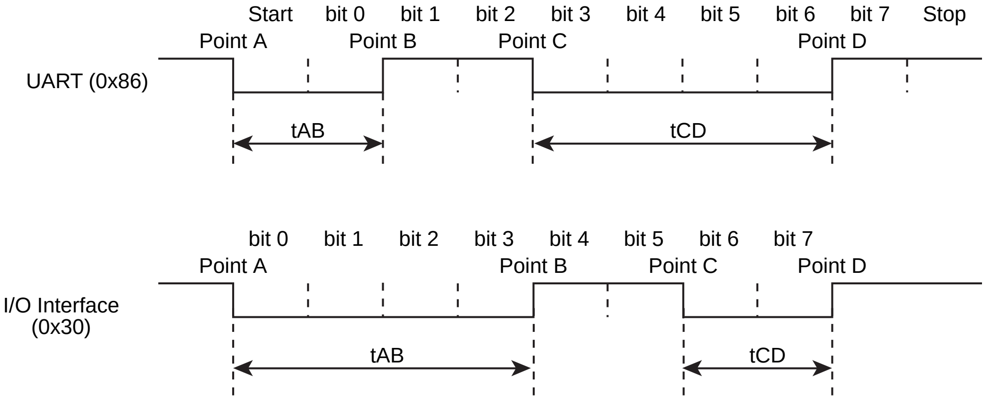


Figure 22-5 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 22-6 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts when each time the logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals tend to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode may have this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 of the desired baud rate.

The flowchart in Figure 22-6 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Note that if the baud rate is too high or the timer operating frequency is too low, each timer value becomes small. It causes an unintentional behavior of the controller. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

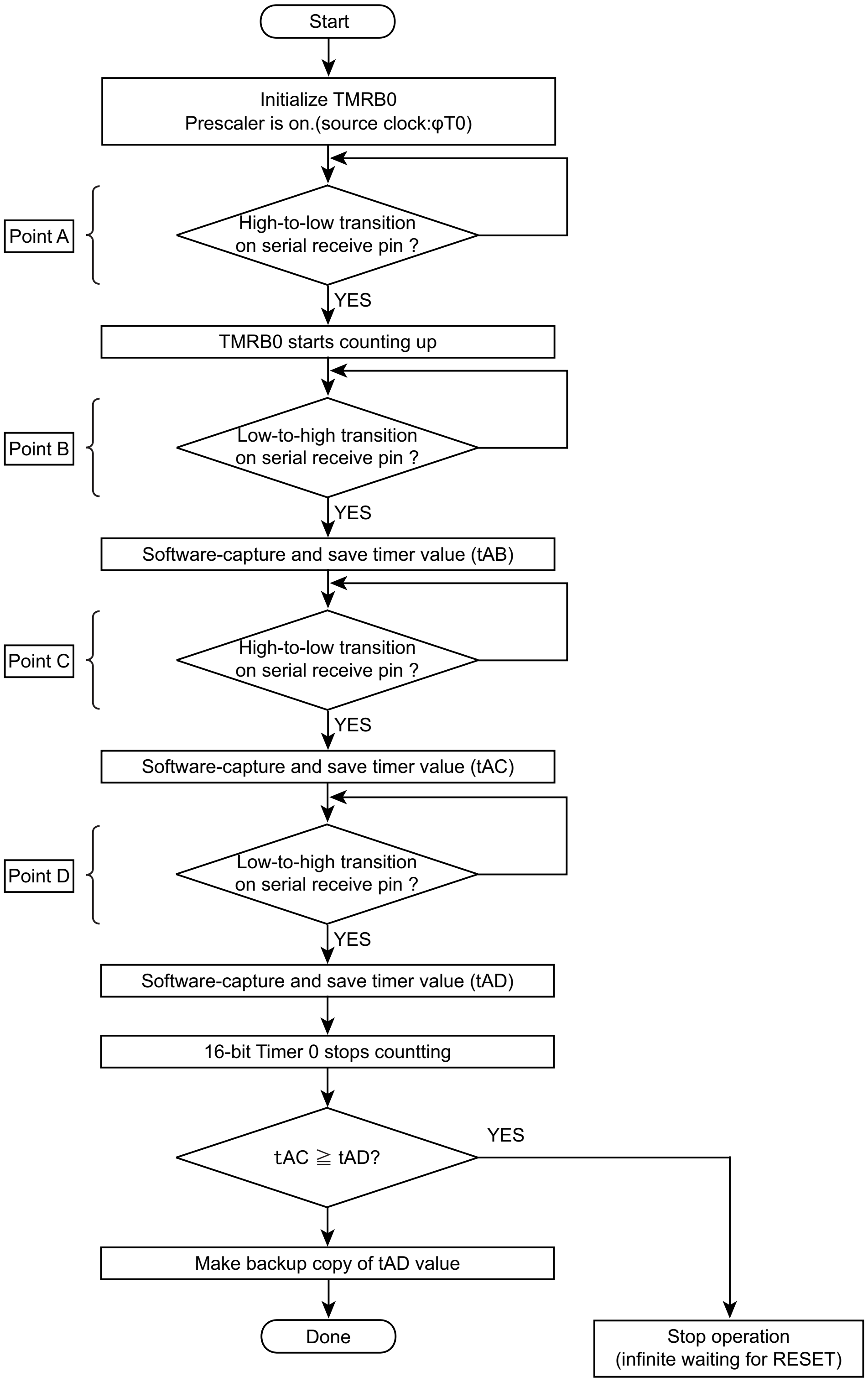


Figure 22-6 Serial Operation Mode Byte Reception Flowchart

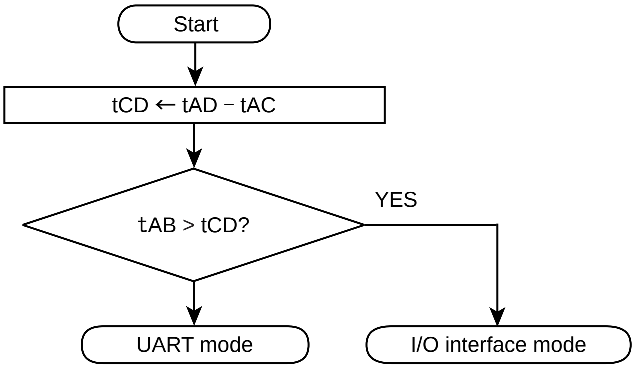


Figure 22-7 Serial Operation Mode Determination Flowchart

22.2.10.5 Password

The RAM Transfer command (0x10) causes the boot program to perform password verification. Following an echo-back of the command code, the boot program verifies the contents of the 12-byte password area within the flash memory. The following table shows the password area of each product.

Product name	Area
TMPM3U6FY/FW	0x3F83_FFF4 to 0x3F83_FFFF

Note: **If a password is set to 0xFF (erased data area), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

If all these address locations contain the same bytes of data other than 0xFF, a password area error occurs as shown in Figure 22-8. In this case, the boot program returns an error acknowledge (0x11) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all 0xFFs.

Receiving data (5th to 16th bytes) from the controller is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

The password verification is performed even if the security function is enabled.

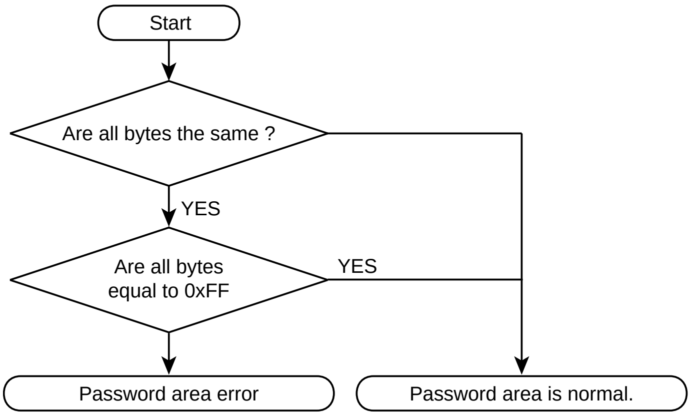


Figure 22-8 Password Area Verification Flowchart

22.2.10.6 Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together with ignoring the carries and calculating the 8-bit two’s complement by using lower 8 bits. The controller must perform the same checksum operation in transmitting checksum bytes.

Example) Assume the Show CheckSum command provides the upper and lower bytes of the sum as 0xE5 and 0xF6. To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together

0xE5 + 0xF6 = 0x1DB

Calculate the two’s complement by using lower 8 bits, and that is the checksum byte. Then send 0x25 to the controller.

0 - 0xDB = 0x25

22.2.11 General Boot Program Flowchart

Figure 22-9 shows an overall flowchart of the boot program.

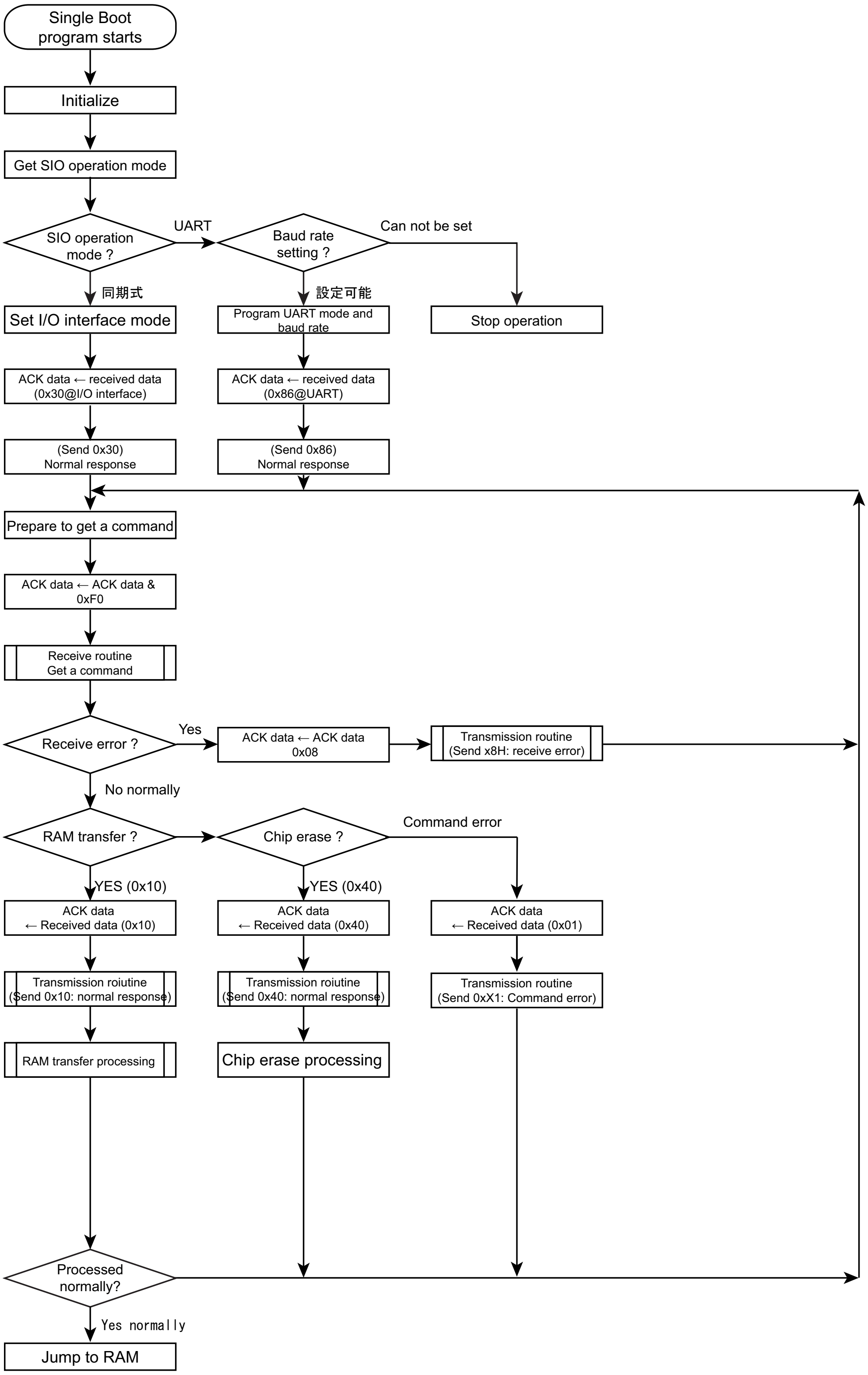

Single Boot program starts
Initialize
Get SIO operation mode
SIO operation mode ?
UART
Baud rate setting ?
Can not be set
同期式
設定可能
Set I/O interface mode
Program UART mode and baud rate
Stop operation
ACK data ← received data (0x30@I/O interface)
ACK data ← received data (0x86@UART)
(Send 0x30) Normal response
(Send 0x86) Normal response
Prepare to get a command
ACK data ← ACK data & 0xF0
Receive routine Get a command
Receive error ?
Yes
ACK data ← ACK data 0x08
Transmission routine (Send x8H: receive error)
No normally
RAM transfer ?
Chip erase ?
Command error
YES (0x10)
YES (0x40)
ACK data ← Received data (0x10)
ACK data ← Received data (0x40)
ACK data ← Received data (0x01)
Transmission roiutine (Send 0x10: normal response)
Transmission roiutine (Send 0x40: normal response)
Transmission routine (Send 0xX1: Command error)
RAM transfer processing
Chip erase processing
Processed normally?
Yes normally
Jump to RAM

Figure 22-9 Overall Boot Program Flowchart

22.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM after shifting to the user boot mode.

22.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands.

In writing or erasing, use 32-bit data transfer command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 22-12 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erase the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	The write or erase operation can be individually inhibited for each block.

22.3.1.1 Block Configuration

(1) TMPM3U6FY/FW

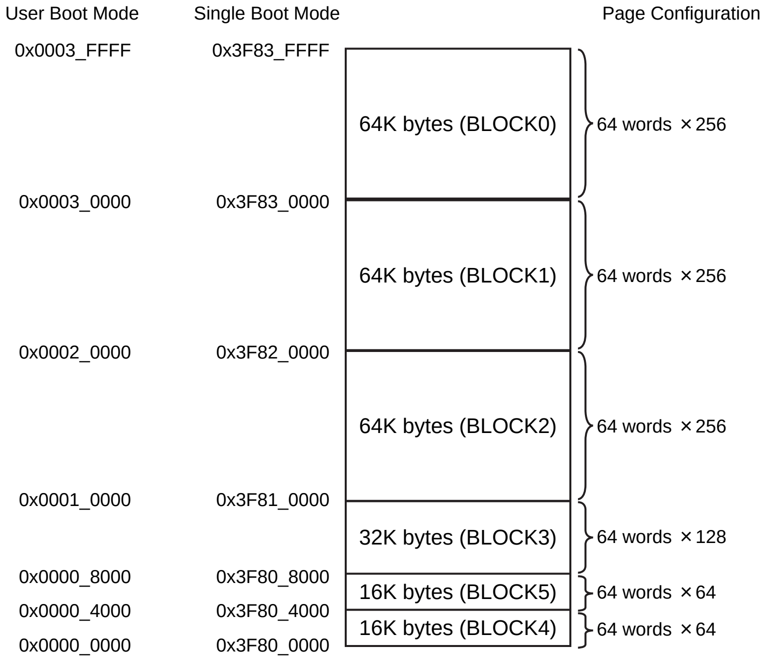


Figure 22-10 Block Configuration of Flash Memory (TMPM3U6FY)

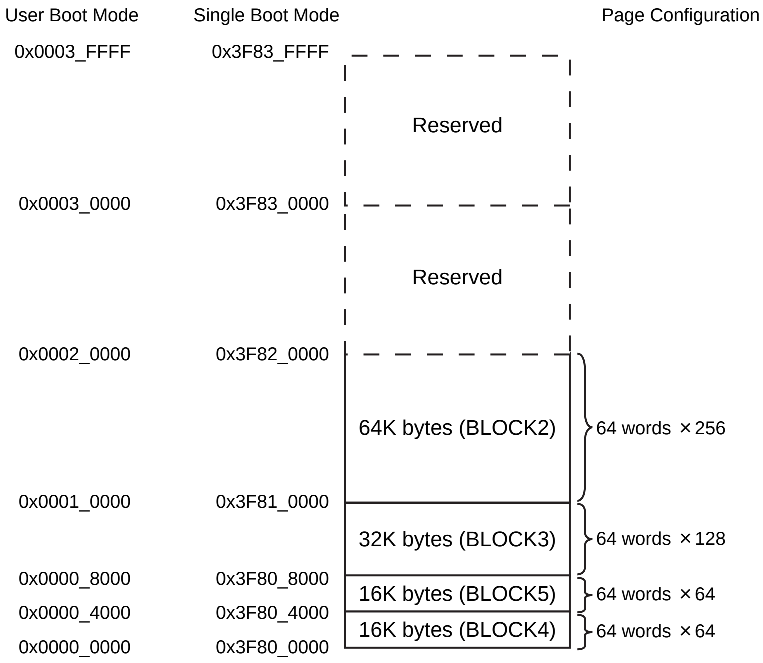


Figure 22-11 Block Configuration of Flash Memory (TMPM3U6FW)

22.3.1.2 Basic Operation

This flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exception other than reset and debug exceptions while a debug port is connected. Any exception generation cannot set the device to the read mode except when a hardware reset is generated.

(1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- Read / reset command and Read command (software reset)

  When ID-Read command is used, the reading operation is terminated instead of automatically returning to the read mode. In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used. The Read command is used to return to the read mode after executing 32-bit data transfer command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.
- With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.

(2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read.

While commands are generally comprised of several bus cycles and the operation applying to the 32-bit (word) data transmission command to the flash memory is called "bus write cycle". The bus write cycles have a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of command write is operated in accordance with a predefined specific order. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

Note 1: **Command sequences are executed from outside the flash memory area.**

Note 2: **Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, do not generate any interrupt (except debug exceptions when a debug port is connected). If such an operation is made, it may result in an unexpected read access to the flash memory, and the command sequencer may not be able to correctly recognize the command. While it may cause an abnormal termination of the command sequence, it also may cause an incorrect recognition of the command.**

Note 3: **For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle where FCFLCS <RDY / BSY> is set to "1". It is recommended to subsequently execute a Read command.**

Note 4: **Upon issuing a command, if any address or data is incorrectly written, be sure to perform a software reset to return to the read mode again.**

22.3.1.3 Reset (Hardware reset)

A hardware reset is used to cancel the operational mode set by the command write operation when forcibly terminated during auto programming/erasing or abnormal termination in the automatic operation.

The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the $\overline{\text{RESET}}$ input pin of this device is set to VIL or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section "22.2.1 Reset Operation" for CPU reset operations. After a given reset input, the CPU will read the reset vector data from the flash memory and starts operation after the reset is removed.

22.3.1.4 Commands

(1) Automatic Page Program

Writing to a flash memory device is to change "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For changing "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM3U6FY/FW contains 64 words in a page. A 64 word block is defined by the same [31:8] address. It starts from the address [7:0] = 0x00 and ends at the address [7:0] = 0xFF. This programming unit is hereafter referred to as a "page".

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by FCFLCS [0] <RDY/BSY>.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more. Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content may cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. After the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at one time). Be sure to use the 32-bit data transfer command in writing commands after the fourth bus cycle. At this time, any 32-bit data transfer commands shall not be placed across word boundary. After the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0". For example, if the top address of a page is not to be written, set the input data in the fourth bus write cycle to 0xFFFFFFFF as a command write.

Once the third bus cycle is executed, the automatic page programming is in operation. This condition can be checked by monitoring FCFLCS<RDY / BSY>. Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted. When a single page has been command written with normally terminating the automatic page writing process, FCFLCS<RDY / BSY> is set to "1" then it returns to the read mode.

When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FCFLCS<RDY/

BSY>. If automatic programming has failed, the flash memory is locked in the current mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

Note: **Software reset becomes ineffective after the fourth bus write cycle of the automatic page programming command.**

(2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FCFLCS<RDY / BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected block cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the current mode and will not return to the read mode.

For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

(3) Automatic block erase (for each block)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FCFLCS <RDY / BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected block cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

(4) Automatic programming of protection bits (for each block)

This device is implemented with protection bits. This protection can be set for each block. See Table 22-18 for table of protection bit addresses. This device assigns 1 bit to 1 block as a protection bit. The applicable protection bit is specified by PBA in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY/BSY>. Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, all FCFLCS <BLPRO> are set to "1" indicating that it is in the protected state. This disables subsequent writing and erasing of all blocks.

Note: **Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. FCFLCS <RDY/BSY> turns to "0" after entering the seventh bus write cycle.**

(5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on whether all <BLPRO> in the FCFLCS register are set to "1" or not, when FCSECBIT<FCSECBIT> is set to "1". Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See Chapter "Protect/security function" for details.

- When all the FCFLCS <BLPRO> are set to "1" (all the protection bits are programmed):

  When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FCFLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FCFLCS will be set to "0x00000001". Since no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FCFLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as required.

- When FCFLCS <BLPRO> includes "0" (not all the protection bits are programmed):

  If the automatic protection bit is cleared to "0", the protection condition is canceled. With this device, protection bits can be programmed to an individual block and performed bit-erase operation in the four bits unit as shown in Table 22-18. The target bits are specified in the seventh bus write cycle.The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FCFLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FCFLCS <BLPRO> selected for erasure are set to "0".

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

Note: **The FCFLCS <RDY / BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.**

(6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). After the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repeatedly executed. For returning to the read mode, use the Read/reset command or hardware reset command.

22.3.1.5 Flash control / status register

Base Address = 0x41FF_F000

Register name		Address (Base+)
Reserved	-	0x0000, 0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024, 0x0028
Reserved	-	0x0040, 0x0044
Reserved	-	0x0050, 0x0058
Reserved	-	0x0060 - 0x00B8

Note: **Do not access to the reserved address.**

(1) FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block 5 to 0 0: disabled 1: enabled Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1", it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15-1	−	R	Read as 0.
0	RDY/BSY	R	Ready / Busy (Note 1) 0:Auto operating 1:Auto operation terminated. Ready/Busy flag bit The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1".

Note 1: **This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 μs regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

Note 2: **The value varies depending on protection applied.**

(2) FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bits 0:disabled 1:enabled

Note: **This register is initialized by Power on reset.**

22.3.1.6 List of Command Sequences

Table 22-13 shows the address and the data of each command of flash memory.

Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by 32-bit (word) data transfer commands. (In the following table, only lower 8 bits data are shown.)

See Table 22-14 for the detail of the address bit configuration. Use a value of "Addr." in the Table 22-13 for the address [15:8] of the normal command in the Table 22-14.

Note: **Always set "0" to the address bits [1:0] in the entire bus cycle.**

Table 22-13 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read / Reset	0x54XX	0xAAXX	0x54XX	RA	–	–	–
	0xAA	0x55	0xF0	RD	–	–	–
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page programming	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Auto block erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Protection bit programming	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address
  PD: Program data (32 bit data)

  After fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

22.3.2 Address bit configuration for bus write cycles

Table 22-14 is used in conjunction with "Table 22-13 Flash Memory Access from the Internal CPU".

Address setting can be performed according to the normal bus write cycle address configuration from the first bus cycle. "0" is recommended in the Table 22-14 Address Bit Configuration for Bus Write Cycles can be changed as necessary.

Table 22-14 Address Bit Configuration for Bus Write Cycles

<table>
<tr><th>Address</th><th>Addr
[31:19]</th><th>Addr
[18]</th><th>Addr
[17]</th><th>Addr
[16]</th><th>Addr
[15]</th><th>Addr
[14]</th><th>Addr
[13:11]</th><th>Addr
[10]</th><th>Addr
[9]</th><th>Addr
[8]</th><th>Addr
[7:0]</th></tr>
<tr><td rowspan="2">Normal commands</td><td colspan="11">Normal bus write cycle address configuration</td></tr>
<tr><td>Flash area</td><td colspan="3">"0" is recommended.</td><td colspan="6">Command</td><td>Addr[1:0] = "0" (fixed)
Others:0 (recommended)</td></tr>
<tr><td rowspan="2">ID-READ</td><td colspan="11">IA: ID address (Set the fourth bus write cycle address for ID-Read operation)</td></tr>
<tr><td>Flash area</td><td colspan="3">"0" is recommended.</td><td colspan="2">ID address</td><td colspan="5">Addr[1:0] = "0" (fixed), Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Block erase</td><td colspan="11">BA: Block address (Set the sixth bus write cycle address for block erase operation)</td></tr>
<tr><td colspan="5">Block selection (Table 22-15/Table 22-16)</td><td colspan="6">Addr[1:0] = "0" (fixed), Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Auto page programming</td><td colspan="11">PA: Program page address (Set the fourth bus write cycle address for page programming operation)</td></tr>
<tr><td colspan="9">Page selection</td><td colspan="2">Addr[1:0] = "0" (fixed)
Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Protection bit programming</td><td colspan="11">PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming)</td></tr>
<tr><td>Flash area</td><td colspan="2">Protection bit selection
(Table 22-17)</td><td colspan="5">Fixed to "0".</td><td colspan="2">Protect bit selection
(Table 22-17)</td><td>Addr[1:0] = "0" (fixed)
Others:0 (recommended)</td></tr>
<tr><td rowspan="2">Protection bit erase</td><td colspan="11">PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)</td></tr>
<tr><td>Flash area</td><td colspan="2">Protection bit selection
(Table 22-18)</td><td colspan="7">Fixed to "0".</td><td>Addr[1:0] = "0" (fixed)
Others:0 (recommended)</td></tr>
</table>

As block address, specify any address in the block to be erased.

Table 22-15 Block Address Table (TMPM3U6FY)

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
4	0x0000_0000 to 0x0000_3FFF	0x3F80_0000 to 0x3F80_3FFF	16
5	0x0000_4000 to 0x0000_7FFF	0x3F80_4000 to 0x3F80_7FFF	16
3	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
2	0x0001_0000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	64
1	0x0002_0000 to 0x0002_FFFF	0x3F82_0000 to 0x3F82_FFFF	64
0	0x0003_0000 to 0x0003_FFFF	0x3F83_0000 to 0x3F83_FFFF	64

Note: **As for the addresses from the first to the fifth bus cycles, specify the upper addresses of the blocks to be erased.**

Table 22-16 Block Address Table (TMPM3U6FW)

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
4	0x0000_0000 to 0x0000_3FFF	0x3F80_0000 to 0x3F80_3FFF	16
5	0x0000_4000 to 0x0000_7FFF	0x3F80_4000 to 0x3F80_7FFF	16
3	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
2	0x0001_0000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	64
1	Reserved	Reserved	64
0	Reserved	Reserved	64

Note 1: **Do not access the reserved area.**

Note 2: **As for the addresses from the first to the fifth bus cycles, specify the upper addresses of the blocks to be erased.**

Table 22-17 Protection Bit Programming Address Table (TMPM3U6FY/FW)

Block	Protection bit	The seventh bus write cycle address				
		Address [18]	Address [17]	Address [16:11]	Address [10]	Address [9]
Block0	<BLPRO[0]>	0	0	Fixed to "0".	0	0
Block1	<BLPRO[1]>	0	0		0	1
Block2	<BLPRO[2]>	0	0		1	0
Block3	<BLPRO[3]>	0	0		1	1
Block4	<BLPRO[4]>	0	1		0	0
Block5	<BLPRO[5]>	0	1		0	1

Table 22-18 Protection Bit Erase Address Table

Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address [18]	Address [17]
Block0 to 3	<BLPRO[0:3]>	0	0
Block4 to 5	<BLPRO[4:5]>	0	1

Note: **The protection bit erase command cannot erase by individual block.**

Table 22-19 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following 32-bit data transfer command (ID)

IA[15:14]	ID[7:0]	Code
00	0x98	Manufacturer code
01	0x5A	Device code
10	Reserved	−
11	0x13	Macro code

22.3.2.1 Flowchart

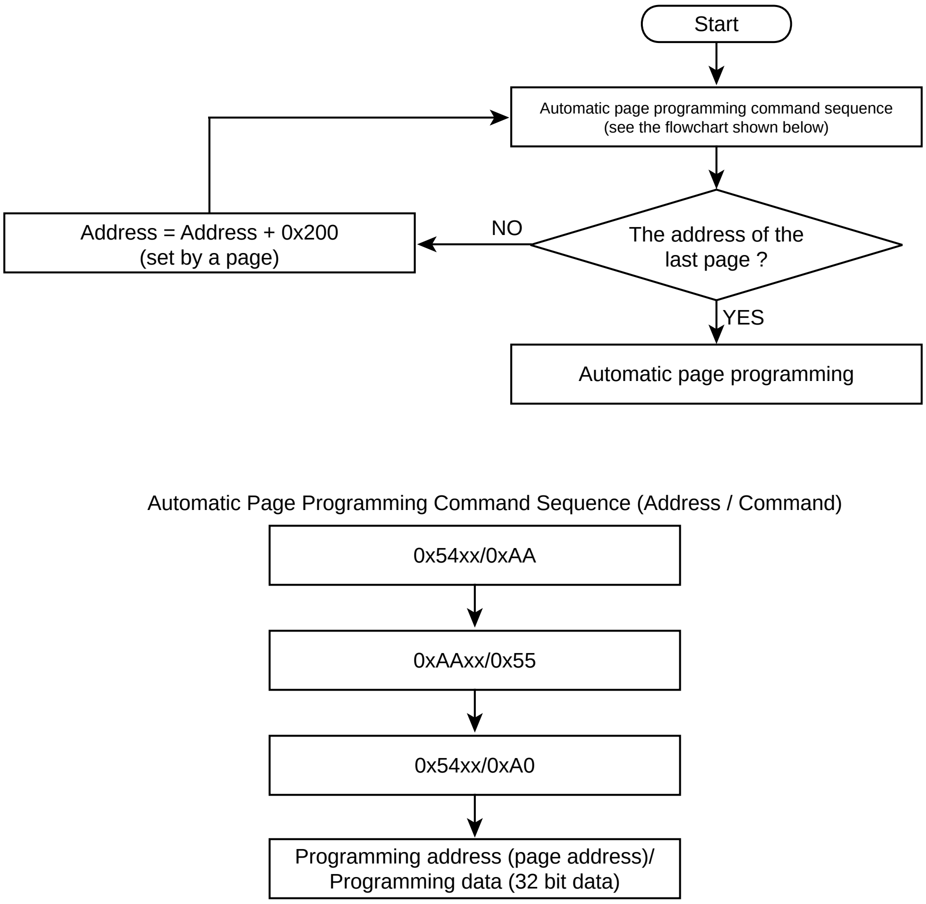

Start
Automatic page programming command sequence
(see the flowchart shown below)
The address of the last page ?
NO
Address = Address + 0x200
(set by a page)
YES
Automatic page programming
Automatic Page Programming Command Sequence (Address / Command)
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0xA0
Programming address (page address)/
Programming data (32 bit data)

Figure 22-12 Automatic Programming

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

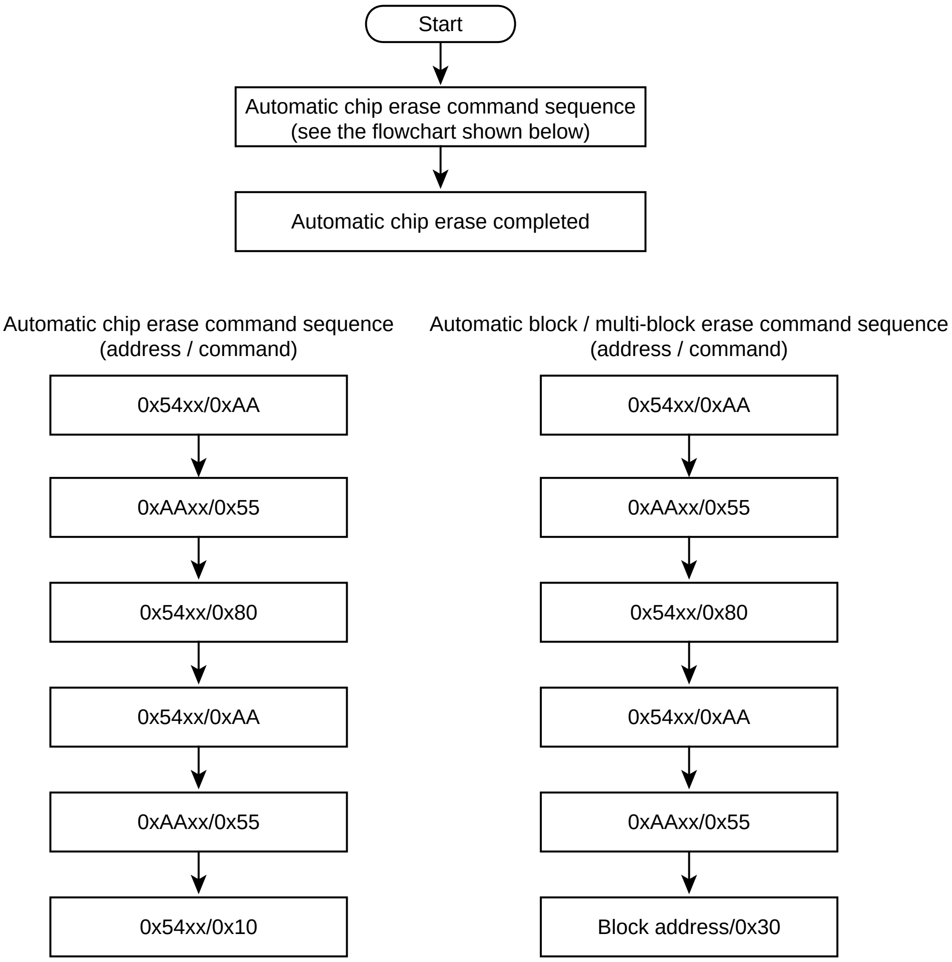

Start
Automatic chip erase command sequence (see the flowchart shown below)
Automatic chip erase completed
Automatic chip erase command sequence (address / command)
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0x80
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0x10
Automatic block / multi-block erase command sequence (address / command)
0x54xx/0xAA
0xAAxx/0x55
0x54xx/0x80
0x54xx/0xAA
0xAAxx/0x55
Block address/0x30

Figure 22-13 Automatic Erase

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

23. ROM protection

23.1 Outline

The TMPM3U6FY/FW offers two kinds of ROM protection/ security functions.

One is a write/ erase-protection function for the internal flash ROM data.

The other is a security function that restricts internal flash ROM data readout and debugging.

23.2 Features

23.2.1 Write/ erase-protection function

The write/ erase-protection function enables the internal flash to prohibit the writing and erasing operation for each block.

To activate the function, write "1" to the corresponding bits to a block to protect. Writing "0" to the bits cancels the protection.

The protection settings of the bits can be monitored by the FCFLCS <BLPRO[5:0]> bit. See the chapter "Flash" for programming details.

23.2.2 Security function

The security function restricts flash ROM data readout and debugging.

This function is available under the conditions shown below.

1. The FCSECBIT <SECBIT> bit is set to"1".
2. All the protection bits (the FCFLCS<BLPRO> bits) used for the write/erase-protection function are set to "1".

Note: **The FCSECBIT <SECBIT> bit is set to "1" at a power-on reset right after power-on.**

Table 23-1 shows details of the restrictions by the security function.

Table 23-1 Restrictions by the security function

Item	Details
1) ROM data readout	Data can be read from CPU.
2) Debug port	Communication of JTAG/SW and trace are prohibited
3) Command for flash memory	Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/ erase-protection erases all the protection bits.

23.3 Register

Base Address = 0x41FF_F000

Register name		Address(Base+)
Reserved	-	0x0000,0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024,0x0028
Reserved	-	0x0040,0x0044
Reserved	-	0x0050,0x0058
Reserved	-	0x0060 − 0x00B8

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

23.3.1 FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY/BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	−	R	Read as 0.
21-16	BLPRO5 to BLPRO0	R	Protection for Block5 to 0 0: disabled 1: enabled Protection status bits Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15-1	−	R	Read as 0.
0	RDY/BSY	R	Ready/Busy (Note 1) 0: Auto operating 1:Auto operation terminated Ready/Busy flag bit The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Note 1: **This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 ms regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

Note 2: **The value varies depending on protection applied.**

23.3.2 FCSECBIT(Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as 0.
0	SECBIT	R/W	Security bit 0: Disabled 1: Enabled

Note: This register is initialized by cold reset.

23.4 Writing and erasing

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.

23.4.1 Protection bits

Writing to the protection bits is done on block-by-block basis.

When the settings for all the blocks are "1", erasing must be done after setting the FCSECBIT <SECBIT> bit to "0". Setting "1" at that situation erases all the protection bits. To write and erase the protection bits, command sequence is used.

See the capter "Flash" for details

23.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on.

The bit is rewritten by the following procedure.

1. Write the code 0xa74a9d23 to FCSECBIT register.
2. Write data within 16 clocks from the above.1.

Note: The above procedure is enabled only when using 32-bit data transfer command.

24. Debug Interface

24.1 Specification Overview

TMPM3U6FY/FW contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the debug-ging tools and the Embedded Trace Macrocell™(ETM) unit for instruction trace output.Trace data is output to the dedicated pins(TRACEDATA[1:0], SWV) for the debugging via the on-chip Trace Port Interface Unit (TPIU).

For details about SWJ-DP, ETM and TPIU, refer to the Arm manual "Cortex-M3 Technical Reference Manual" .

24.2 SWJ-DP

SWJ-DP supports the Serial Wire Debug Port (SWCLK, SWDIO) and the JTAG Debug Port (TDI, TDO, TMS, TCK, $\overline{\text{TRST}}$).

24.3 ETM

ETM supports two data signal pins (TRACEDATA[1:0]), one clock signal pin (TRACECLK) and trace output from Serial Wire Viewer (SWV).

24.4 Pin Functions

The debug interface pins can also be used as general-purpose ports.

The PB3 and PB4 pins are shared between the JTAG debug port function and the Serial Wire Debug Port function. The PB5 pin is shared between the JTAG debug port function and the SWV trace output function.

Table 24-1 SWJ-DP,ETM Debug Functions

SWJ-DP Pin Name	General-purpose Port Name	JTAG Debug Function		SW Debug Function	
		I / O	Explanation	I / O	Explanation
TMS / SWDIO	PB3	Input	JTAG Test Mode Selection	I / O	Serial Wire Data Input/Output
TCK / SWCLK	PB4	Input	JTAG Test Check	Input	Serial Wire Clock
TDO / SWV	PB5	Output	JTAG Test Data Output	(Output)(Note)	(Serial Wire Viewer Output)
TDI	PB6	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PB7	Input	JTAG Test $\overline{\text{RESET}}$	-	-
TRACECLK	PB0	Output	TRACE Clock Output		
TRACEDATA0	PB1	Output	TRACE DATA Output0		
TRACEDATA1	PB2	Output	TRACE DATA Output1		

Note: **When SWV function is enabled.**

After reset, PB3, PB4, PB5, PB6 and PB7 pins are configured as debug port function pins. The functions of other debug interface pins need to be programmed as required.

When using a low power consumption mode, take note of the following points.

Note 1: If PB3 and PB5 are configured as TMS/SWDIO and TDO/SWV, output continues to be enabled even in STOP mode regardless of the setting of the CGSTBYCR<DRVE> bit.

Note 2: If PB4 is configured as a debug function pin, it prevents a low power consumption mode from being fully effective. Configure PB4 to function as a general-purpose port if the debug function is not used.

Table 24-2 summarizes the debug interface pin and related port settings after reset.

Table 24-2 Debug Interface Pins and Related Port Settings after Reset

Port Name (Bit Name)	Debug Function	Value of Related port settings after reset				
		Function (PxFR)	Input (PxIE)	Output (PxCR)	Pull-up (PxPUP)	Pull-down (PxPDN)
PB3	TMS/SWDIO	1	1	1	1	0
PB4	TCK/SWCLK	1	1	0	0	1
PB5	TDO/SWV	1	0	1	0	0
PB6	TDI	1	1	0	1	0
PB7	$\overline{\text{TRST}}$	1	1	0	1	0
PB0	TRACECLK	0	0	0	0	0
PB1	TRACEDATA0	0	0	0	0	0
PB2	TRACEDATA1	0	0	0	0	0

- : Don’t care

24.5 Peripheral Functions in Halt Mode

When the Cortex-M3 core enters in the halt mode, the watchdog-timer (WDT) automatically stops. Other peripheral functions continue to operate.

24.6 Connection with a Debug Tool

24.6.1 About connection with debug tool

Concerning a connection with debug tools, refer to manufactures recommendations.

Debug interface pins contain a pull-up resistor and a pull-down resistor.When debug interface pins are connected with external pull-up or pull-down, please pay attention to input level.

24.6.2 Important points of using debug interface pins used as general-purpose ports

When setting a debugging interface terminal to a general-purpose port by a user's program after reset release,after that the control from a debugging tool is impossible.

Please note that it is necessary to prepare for the structure which changes the general-purpose port to the debugging interface function by some kind of methods to connect a debugging tool again..

Table 24-3 Example Table of using debug interface pins

	Debug interface pins						
	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA[1:0]	TRACE CLK
JTAG+SW (After reset)	o	o	o	o	o	×	×
JTAG+SW (without TRST)	(Note)	o	o	o	o	×	×
JTAG+TRACE	o	o	o	o	o	o	o
SW	×	×	×	o	o	×	×
SW+SWV	×	×	o	o	o	×	×
Debugging function disabled	×	×	×	×	×	×	×

o : Enabled × : Disabled (Usable as general-purpose port)

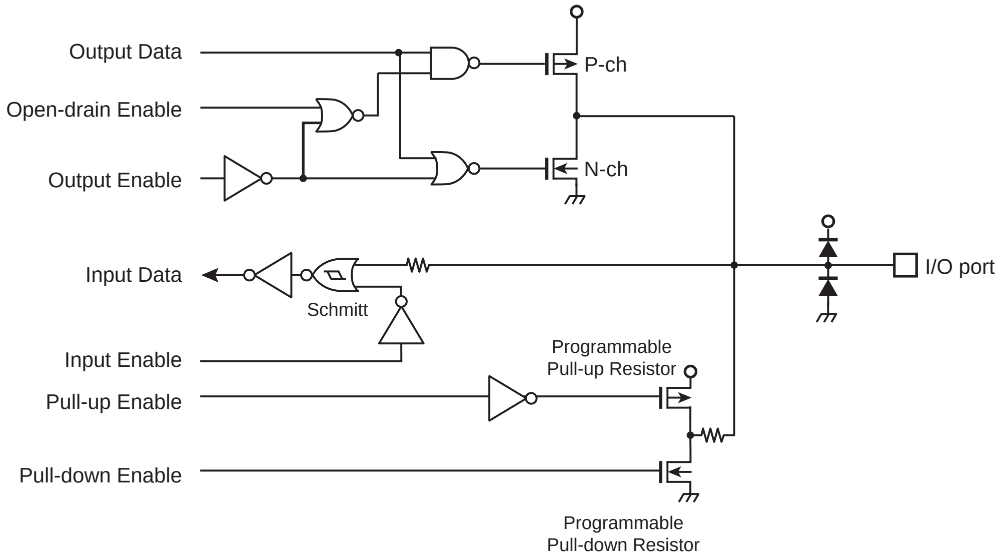
Note: For the treatment of the pin of which the $\overline{\text{TRST}}$ function is assigned, select the $\overline{\text{TRST}}$ function with the function register and set the pin to OPEN or "High level".

25. Port Section Equivalent Circuit Schematic

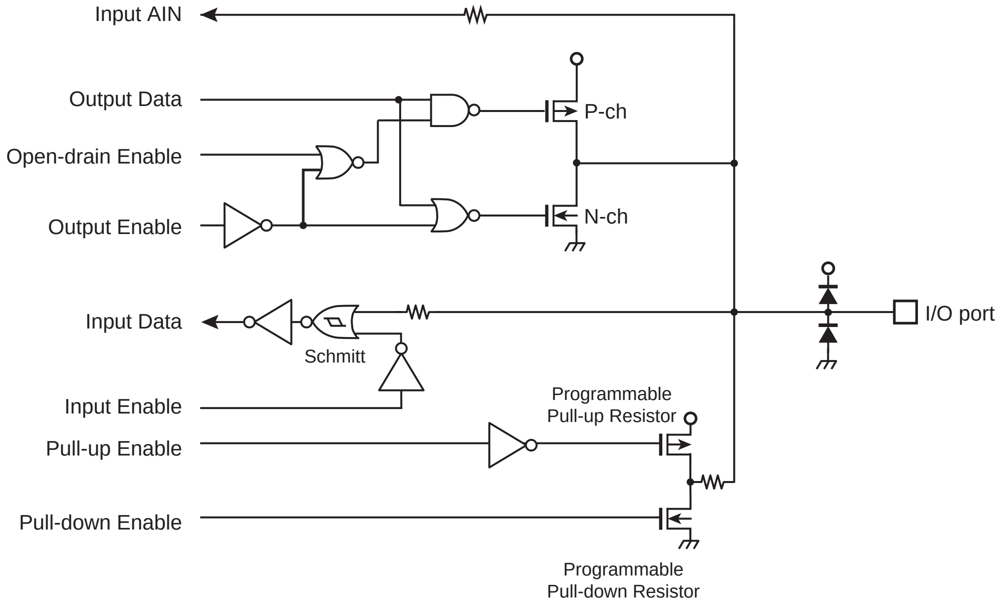
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of Ω to several hundred Ω. Feedback resistor and Damping resistor are shown with a typical value.

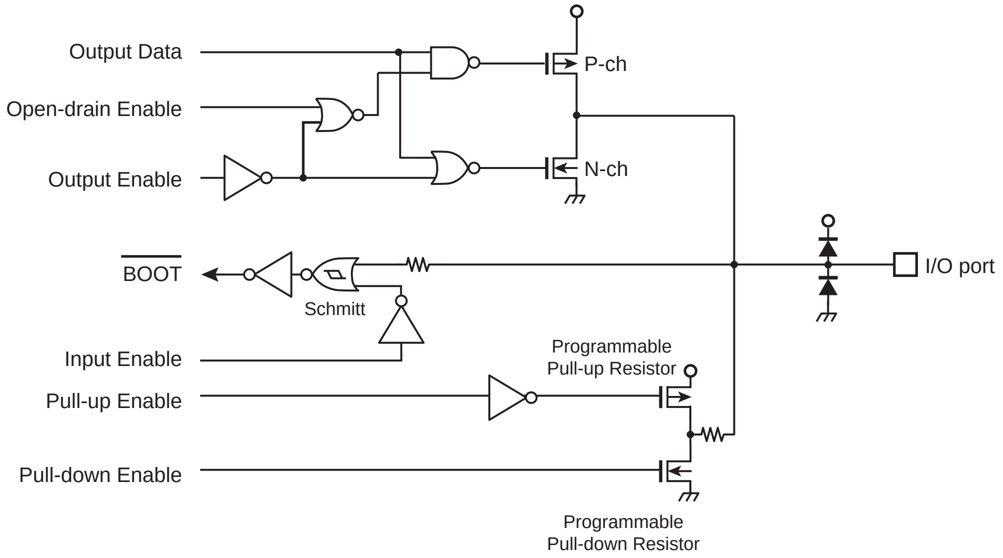
25.1 PA0 to 7, PB0 to 7, PC0 to 7, PD0 to 6, PE0 to 7, PF0 to 4, PG0 to 7, PL2, PN0 to 7



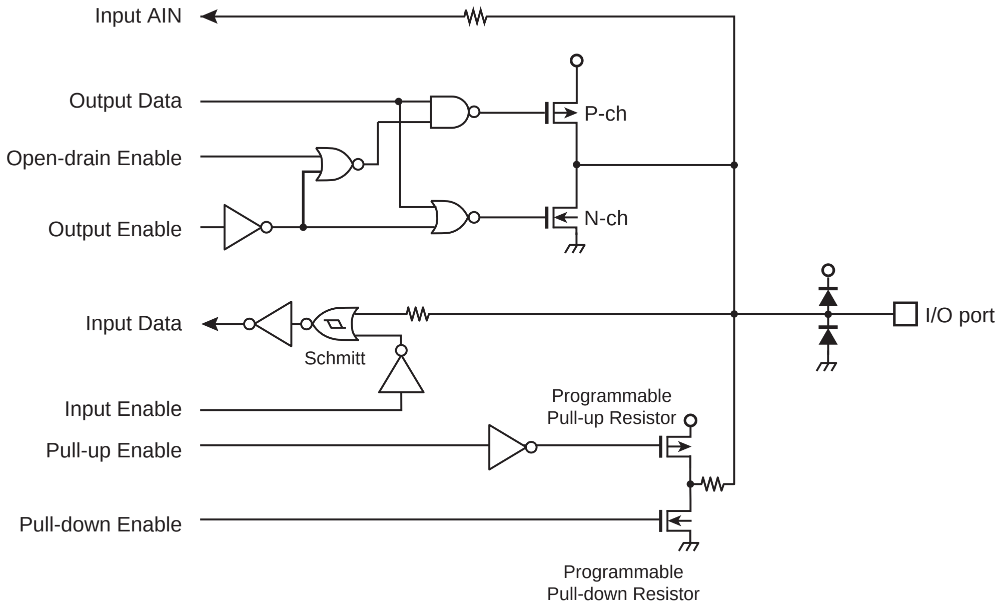
25.2 PH0 to 7, PI0 to 1, PJ0 to 7



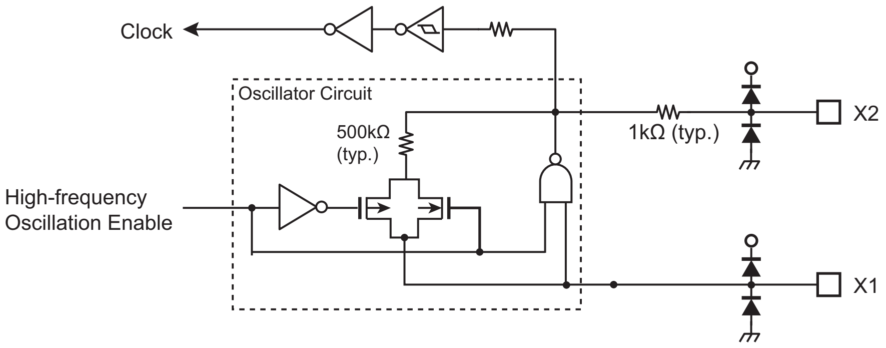
25.3 PL0



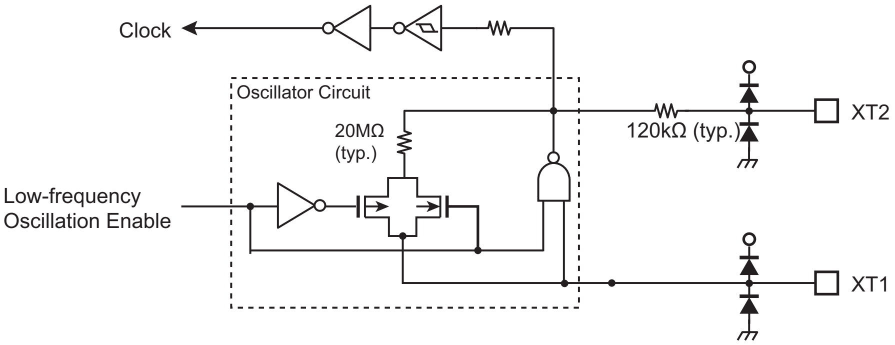
25.4 PM0 to 1, PP0 to 1



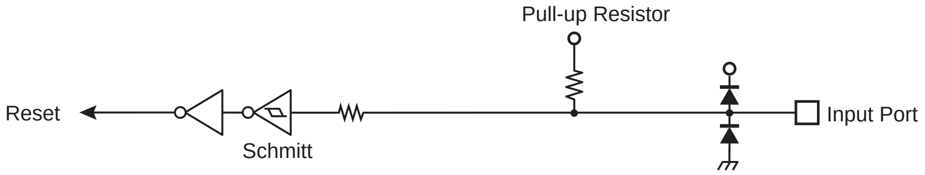
25.5 X1, X2



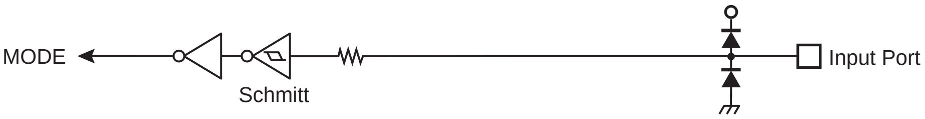
25.6 XT1, XT2



25.7 RESET

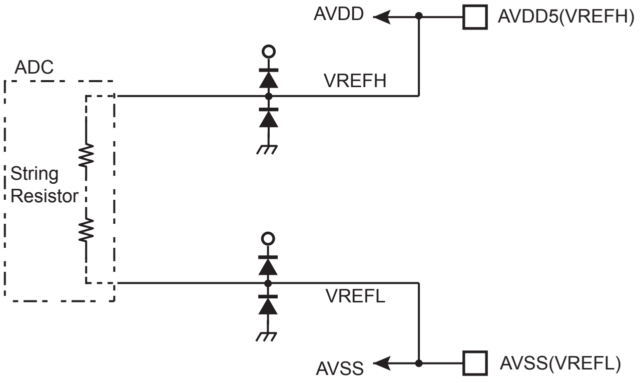


25.8 MODE



(Note)MODE pin is fixed to GND.

25.9 VREFH,VREFL



26. Electrical Characteristics

26.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5	−0.3 to 6.0	V
		RVDD5	−0.3 to 6.0	
		AVDD5	−0.3 to 6.0	
Input voltage		V_{IN}	−0.3 to VDD + 0.3	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	−50	
Power consumption (Ta = 85 °C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−55 to 125	°C
Operating Temperature	Except the following condition	T_{OPR}	−40 to 85	°C
	During Flash W/E and Debugging		0 to 70	

Note: **Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.**

26.2 DC Electrical Characteristics (1/3)

DVDD5 = RVDD5 = AVDD5 = 4.0V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Supply voltage (Note 2)		DVDD5 AVDD5 RVDD5	f_{OSC} = 8 to 10 MHz fsys = 1 to 40 MHz fs = 30 to 34 kHz	4.0	−	5.5	V
Capacitance for VOUT3 (Note 3)		C_{Out}	DVDD5 = 4.0 to 5.5V	3.3	−	4.7	μF
Low-level Input voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{IL1}	4.0 V ≤ DVDD5 ≤ 5.5 V	−0.3	−	0.25 DVDD5	V
	PORT H,I,J	V_{IL2}	4.0 V ≤ AVDD5 ≤ 5.5 V	−0.3	−	0.25 AVDD5	
High-level Input voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{IH1}	4.0 V ≤ DVDD5 ≤ 5.5 V	0.75 DVDD5	−	DVDD5 + 0.3	V
	PORT H,I,J	V_{IH2}	4.0 V ≤ AVDD5 ≤ 5.5 V	0.75 AVDD5	−	AVDD5 + 0.3	
Low-level Output voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{OL1}	DVDD5 ≥ 4.0V I_{OL} = 1.6 mA	−	−	0.4	V
	PORT H,I,J	V_{OL2}	AVDD5 ≥ 4.0V I_{OL} = 1.6 mA	−	−	0.4	
High-level Output voltage	PORT A,B,C,D,E,F,G,L,M,N,P	V_{OH1}	DVDD5 ≥ 4.0V I_{OH} = −1.6 mA	DVDD5 − 0.4	−	−	V
	PORT H,I,J	V_{OH2}	AVDD5 ≥ 4.0V I_{OH} = −1.6 mA	AVDD5 − 0.4	−	−	
Input leakage current		I_{LI}	0.0 V ≤ V_{IN} ≤ DVDD5 0.0 V ≤ V_{IN} ≤ AVDD5	−	0.02	±5	μA
Output leakage current		I_{LO}	0.2 V ≤ V_{IN} ≤ DVDD5 -0.2 V 0.2 V ≤ V_{IN} ≤ AVDD5 -0.2 V	−	0.05	±10	
Pull-up resistor at Reset		R_{RST}	4.0 V ≤ DVDD5 ≤ 5.5 V	38.5	50	71.4	kΩ
Schmitt trigger input width		V_{TH}	4.0 V ≤ DVDD5 ≤ 5.5 V 4.0 V ≤ AVDD5 ≤ 5.5 V	0.3	0.6	−	V
Programmable pull-up/pull-down resistor		P_{KH}	4.0 V ≤ DVDD5 ≤ 5.5 V 4.0 V ≤ AVDD5 ≤ 5.5 V	38.5	50	71.4	kΩ
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz	−	−	10	pF

Note 1: Ta = 25 °C, DVDD5= AVDD5 = RVDD5 = 5 V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD5, AVDD5, and RVDD5.

Note 3: VOUT3 pin should be connected to GND via a capacitance.

26.3 DC Electrical Characteristics (2/3)

DVDD5 = RVDD5 = AVDD5 = 4.0V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note1)	Max	Unit
Low-level output current	I_{OL}	Per pin	−	−	2	mA
	ΣI_{OL1}	Per group 4.0 V ≤ DVDD5 ≤ 5.5 V GrL1 = <PA0-7/PE0-5/PG0-7> GrL2 = <PB0-7/PD0-6/PF0-4/PL0> GrL3 = <PC0-7/PM0-1/PP0-1> GrL4 = <PE6-7/PL2/PN0-7>	−	−	20	mA
	ΣI_{OL2}	Per group 4.0 V ≤ AVDD5 ≤ 5.5 V GrL5 = <PH0-7/PI0-1/PJ0-7>	−	−	9	mA
	ΣI_{OL}	Total(all ports)	−	−	30	mA
High-level output current	I_{OH}	Per pin	−	−	-2	mA
	ΣI_{OH1}	Per group 4.0 V ≤ DVDD5 ≤ 5.5 V GrH1 = <PA0-7/PE0-3/PG0-7/PM0-1/PP0-1> GrH2 = <PB0-7/PC0-7/PD0-6/PF0-4/PL0> GrH3 = <PE4-7/PL2/PN0-7>	−	−	-20	mA
	ΣI_{OH2}	Per group 4.0 V ≤ AVDD5 ≤ 5.5 V GrH4 = <PH0-7/PI0-1/PJ0-7>	−	−	-9	mA
	ΣI_{OH}	Total(all ports)	−	−	-30	mA

Note 1: Ta = 25 °C, DVDD5= AVDD5 = RVDD5 = 5 V, unless otherwise noted.

26.4 DC Electrical Characteristics (3/3)

DVDD5 = RVDD5 = AVDD5 = 4.0V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
NORMAL (Note 2)	I_{DD}	fsys = fc = 40 MHz (fosc = 10 MHz)	−	26	35	mA
IDLE (Note 3)			−	20	28	mA
SLOW		fsys = fs = 32.768 kHz	−	440	4500	μA
SLEEP (Note 4)			−	130	1720	μA
STOP		−	−	100	1700	μA

Note 1: Ta = 25 °C, DVDD5 = RVDD5 = AVDD5 = 5 V, unless otherwise noted.

Note 2: Measurement condition of I_{DD} NORMAL :

Execution program: Dhrystone V2.1 (built-in FLASH operation)

All peripheral functions operate excluding A/DC.

Note 3: Measurement condition of I_{DD} IDLE:

CPU is stopped, some of the peripheral is running.

Note 4: Measurement condition of I_{DD} SLEEP:

All peripheral functions stopped.

CPU is stopped, using RTC,RMC peripheral function only .

26.5 12/10-bit AD Converter Electrical Characteristics

DVDD5 = RVDD5 = 4.5V to 5.5V , DVSS = AVSS = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage (+) (Note 1)	AVDD5	AVDD5 = V_{REFH}	DVDD5 − 0.2	–	DVDD5	V
Analog reference voltage	AVSS	AVSS = V_{REFL}	0	–	0	V
Analog input voltage	V_{AIN}	–	AVSS	–	AVDD5	V
Power supply current of analog reference voltage (Note 4)	I_{REF}	I_{REF} ON (During AD conversion)	–	7.5	10.0	mA
		I_{REF} ON (During AD stop)	–	3.5	5	mA
		I_{REF} OFF (During STOP MODE)	–	3	70	μA
INL error	–	12bit mode AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 μF Conversion time ≥ 1.85 μs	–	–	± 9	LSB (Note 2)
DNL error			–	–	+ 6 to −1	
Offset error			–	–	± 5	
Full-scale error			–	–	+ 8 to −2	
Total error			–	–	+ 12 to −8	
INL error	–	10bit mode AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1μF Conversion time ≥ 1.70 μs	–	–	± 3	LSB (Note 3)
DNL error			–	–	± 2	
Offset error			–	–	± 3	
Full-scale error			–	–	± 3	
Total error			–	–	± 4	

Note 1: A/D when using separate power supply for the converter, you must keep this condition.

Note 2: 1LSB = (AVDD5 − AVSS)/4096 [V]

Note 3: 1LSB = (AVDD5 − AVSS)/1024 [V]

Note 4: The relevant pin for I_{REF} is AVDD5, so that the current flowing into AVDD5 is the power supply current AVDD5 + I_{REF} .

Note 5: This is the characteristic in case Only AD converter is operation.

26.6 AC Electrical Characteristics

26.6.1 AC Measurement Condition

The AC characteristics data of this chapter is measured under the following conditions unless otherwise noted.

- Output levels: High = 0.8 × DVDD5, Low = 0.2 × DVDD5
- Input levels: Refer to low-level input voltage and high-level input voltage in "DC Electrical Characteristics".
- Load capacity: CL = 30pF

26.6.2 Serial Channel (SIO/UART)

26.6.2.1 I/O Interface Mode

In the table below, the letter x represents the SIO operation clock SCLK Clock Low width (input) cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCLK input mode

[Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	–	100	–	ns
SCLK Clock Low width (input)	t_{SCL}	4x	–	100	–	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	–	200	–	
Valid Data Input ← SCLK rise or fall (Note 1)	t_{SRD}	30	–	30	–	
SCLK rise or fall → Input Data hold (Note 1)	t_{HSR}	x + 30	–	55	–	

[Output]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	–	120 (Note 3)	–	ns
SCLK Clock Low width (input)	t_{SCL}	4x	–	120 (Note 3)	–	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	–	240	–	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$	–	0 (Note 2)	–	
SCLK rise or fall → Output Data hold (Note 1)	t_{OHS}	$t_{SCY}/2$	–	120	–	

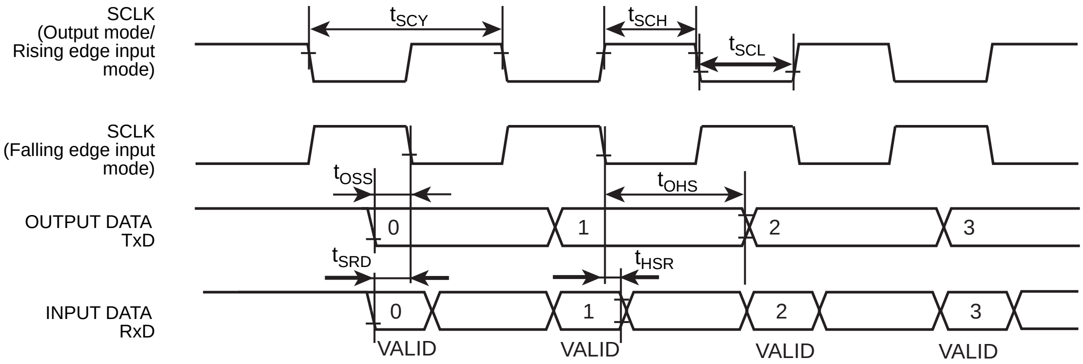
Note 1: SCLK rise/fall: SCLK rise mode uses the rise timing of SCLK. SCLK fall mode uses the fall timing of SCLK.

Note 2: Use the cycle of SCLK in a range where the calculation value keeps positive.

Note 3: The value indicates a minimum value that enables t_{OSS} to be zero or more.

(2) SCLK Output Mode

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	–	100	–	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 20$	–	30	–	
SCLK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 20$	–	30	–	
Valid Data Input ← SCLK rise	t_{SRD}	45	–	45	–	
SCLK rise → Input Data hold	t_{HSR}	0	–	0	–	



26.6.3 Serial Bus Interface (I2C/SIO)

26.6.3.1 I2C Mode

In the table below, the letter x represents the I2C/SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function. It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the <SCK> (SCL output frequency select) field in the SBIxCR.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL Clock frequency	t_{SCL}	0	–	0	100	0	400	kHz
Hold time for START condition	$t_{HD;\ STA}$	–	–	4.0	–	0.6	–	μs
SCL clock low-width (Input) (Note 1)	t_{LOW}	–	–	4.7	–	1.3	–	μs
SCL clock high-width (Input) (Note 2)	t_{HIGH}	–	–	4.0	–	0.6	–	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note 5)	–	4.7	–	0.6	–	μs
Data hold time (Input) (Note 3, 4)	$t_{HD;\ DAT}$	–	–	0.0	–	0.0	–	μs
Data setup time	$t_{SU;\ DAT}$	–	–	250	–	100	–	ns
Setup time for a STOP condition	$t_{SU;\ STO}$	–	–	4.0	–	0.6	–	μs
Bus free time between stop condition and start condition	t_{BUF}	(Note 5)	–	4.7	–	1.3	–	μs

Note 1: SCL clock Low width (output): $(2^{n-1} + 58)/x$

Note 2: SCL clock High width (output): $(2^{n-1} + 14)/x$

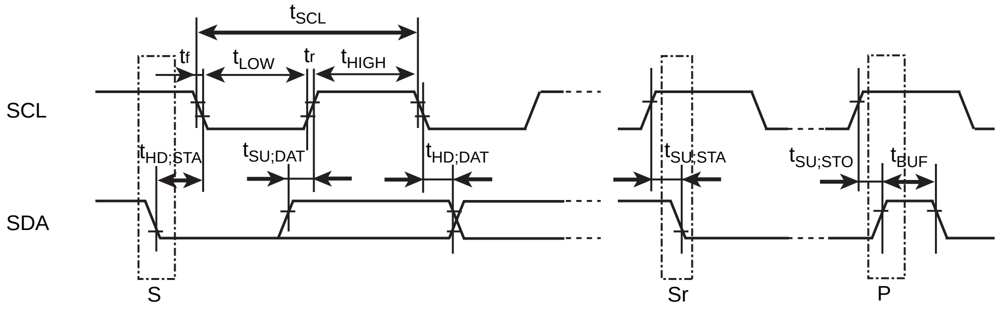
On I2C-bus specification, maximum Speed of Standard Mode/fast mode is 100kHz/400kHz. Internal SCL Frequency setting should comply with fsys and Note1 & Note2 shown above.

Note 3: The output data hold time is equal to 4x of internal SCL.

Note 4: The 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 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

26.6.3.2 Clock-Synchronous 8-bit SIO mode

In the table below, the letter x represents the I2C/SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

(1) SCK input mode (for an SCK signal with a 50% duty cycle)

[Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	–	100	–	ns
SCK Clock Low width (input)	t_{SCL}	4x	–	100	–	
SCK cycle	t_{SCY}	8x	–	200	–	
Valid Data input ← SCK rise	t_{SRD}	30 − x	–	5	–	
SCK rise → Input Data hold	t_{HSR}	2x + 30	–	80	–	

[Output]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	–	120 (Note 2)	–	ns
SCK Clock Low width (input)	t_{SCL}	4x	–	120 (Note 2)	–	
SCK cycle	t_{SCY}	8x	–	240	–	
Output Data ← SCK rise	t_{OSS}	$t_{SCY}/2 - 3x - 45$	–	0 (Note 1)	–	
SCK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 + x$	–	145	–	

Note 1: Keep this value positive by adjusting SCK cycle.

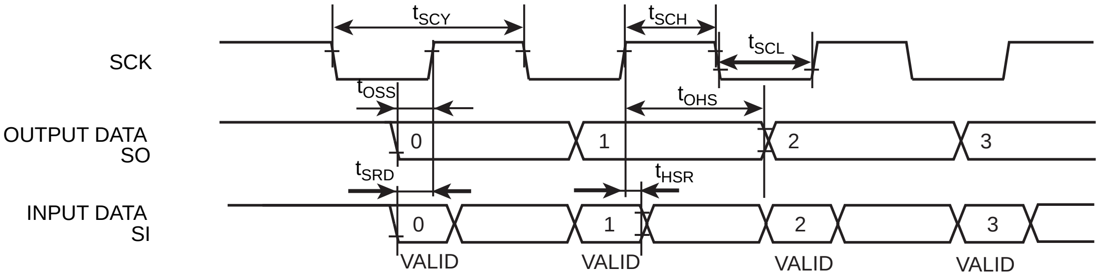
Note 2: The value indicates a minimum value that enables t_{OSS} to be zero or more.

(2) SCK output mode (for an SCK signal with a 50% duty cycle)

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK cycle (programmable)	t_{SCY}	16x (Note1)	−	400	−	ns
Output Data ← SCK rise	t_{OSS}	t_{SCY}/2 − 20 (Note2)	−	180	−	
SCK rise → Output Data hold	t_{OHS}	t_{SCY}/2 − 20	−	180	−	
Valid Data input ← SCK rise	t_{SRD}	x + 45	−	70	−	
SCK rise → Input Data hold	t_{HSR}	0	−	0	−	

Note 1: SCK cycle after automatic wait becomes 14x.

Note 2: t_{OSS} after automatic wait may be t_{SCY}/2−x−20.



26.6.4 Synchronous Serial Interface (SSP)

26.6.4.1 AC Measurement Condition

The letter "T" used in the equations in the table represents the period of the input clock (f_{PCLK}) into the internal prescaler.

- Output levels: High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Input levels: High = 0.9 × DVDD5, Low = 0.1 × DVDD5
- Load capacitance: CL=30pF
- Ta = −40 to 85°C

Note: The "Equation" column in the table shows the specifications under the conditions DVDD5 = 4.0V to 5.5V.

Parameter	Symbol	Equation		fsys=40MHz (m=4,n=12)	Unit
		Min	Max		
SPCLK cycle (master)	T_m	(m)T At least 100ns or more	–	100 (10MHz)	ns
SPCLK cycle (slave)	T_s	(n)T	–	300 (3.3MHz)	
SPCLK rise up time	t_r	–	15	15	
SPCLK fall down time	t_f	–	15	15	
Master mode: SPCLK low-level pulse width	t_{WLM}	(m)T/2 - 20.0	–	30	
Master mode: SPCLK high-level pulse width	t_{WHM}	(m)T/2 - 20.0	–	30	
Slave mode: SPCLK low-level pulse width	t_{WLS}	(n)T/2 - 10.0	–	140	
Slave mode: SPCLK high-level pulse width	t_{WHS}	(n)T/2 - 10.0	–	140	
Master mode: SPCLK rise/fall to output data valid	t_{ODSM}	–	15	15	
Master mode: SPCLK rise/fall to output data hold	t_{ODHM}	(m)T/2 - 15	–	35	
Master mode: SPCLK rise/fall to input data valid delay time	t_{IDSM}	35	–	35	
Master mode: SPCLK rise/fall to input data hold	t_{IDHM}	5	–	5	
Master mode: SPFSS valid to SPCLK rise/fall	t_{OFSM}	(m)T - 15	(m)T + 15	85 to 115	
Slave mode: SPCLK rise/fall to output data valid delay time	t_{ODSS}	–	(3T) + 35	110	
Slave mode: SPCLK rise/fall to Output data hold	t_{ODHS}	(n)T/2 + (2T)	–	200	
Slave mode: SPCLK rise/fall to input data valid delay time	t_{IDSS}	10	–	10	
Slave mode: SPCLK rise/fall to input data hold	t_{IDHS}	(3T) + 15	–	90	
Slave mode: SPFSS valid to SPCLK rise/fall	t_{OFSS}	(n)T - 20	–	280	

Note: Baud rate clock is set under below condition:

- Master mode:

  m = (<CPSDVSR> × (1 + <SCR>)) = fsys/SPCLK

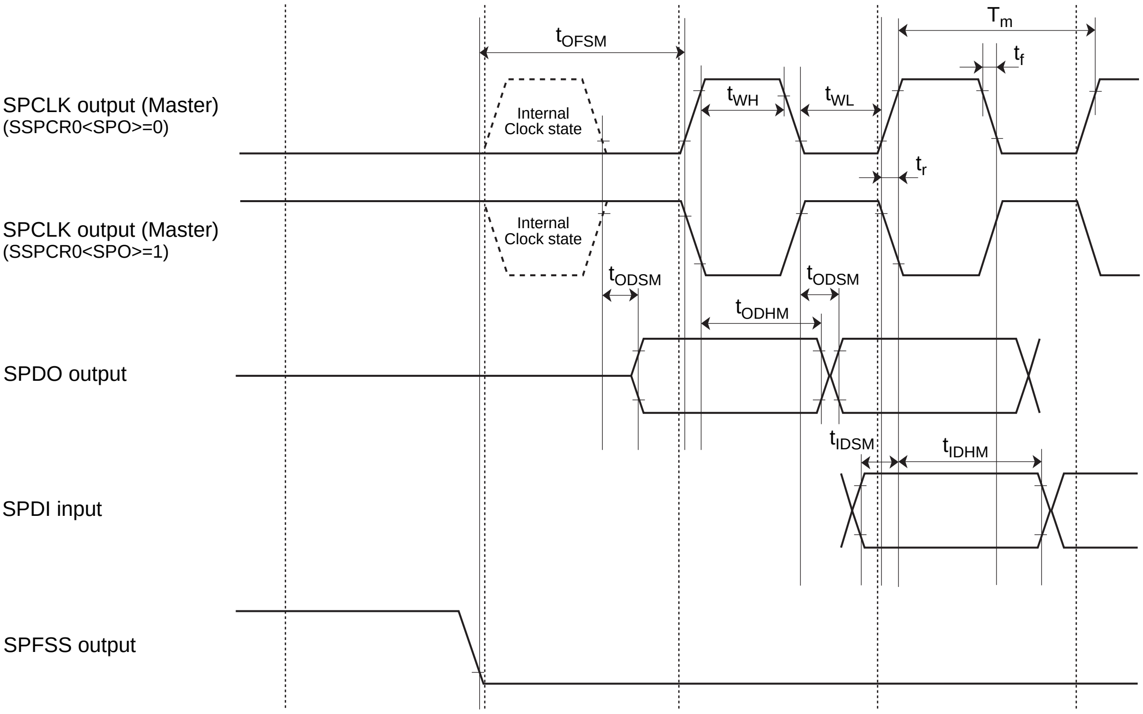
  <CPSDVSR> is set only even number and "m" must set between the range of $65024 \geq m \geq 2$.
- Slave mode

  n = fsys/SPCLK ($65024 \geq n \geq 12$)

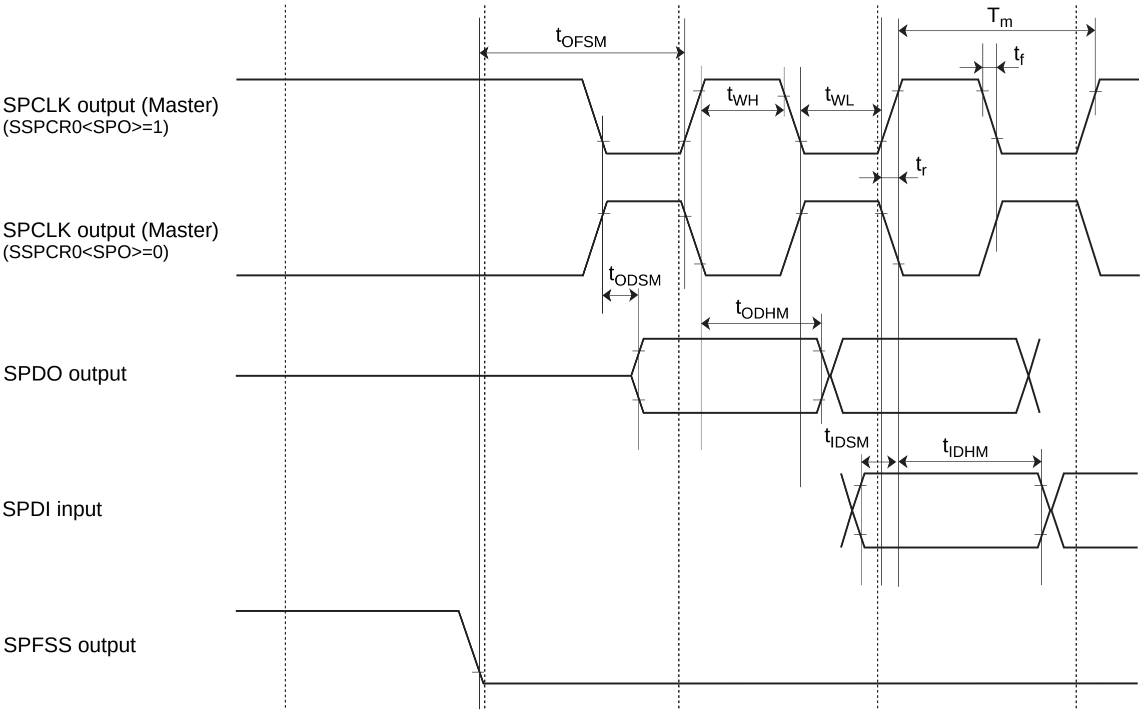
26.6.4.2 SSP SPI mode (Master)

- fsys ≥ 2 × SPxCLK (Maximum)
- fsys ≥ 65024 × SPxCLK (Minimum)

(1) Master SSPCR0<SPH>="0" (Data is latched on the first edge.)



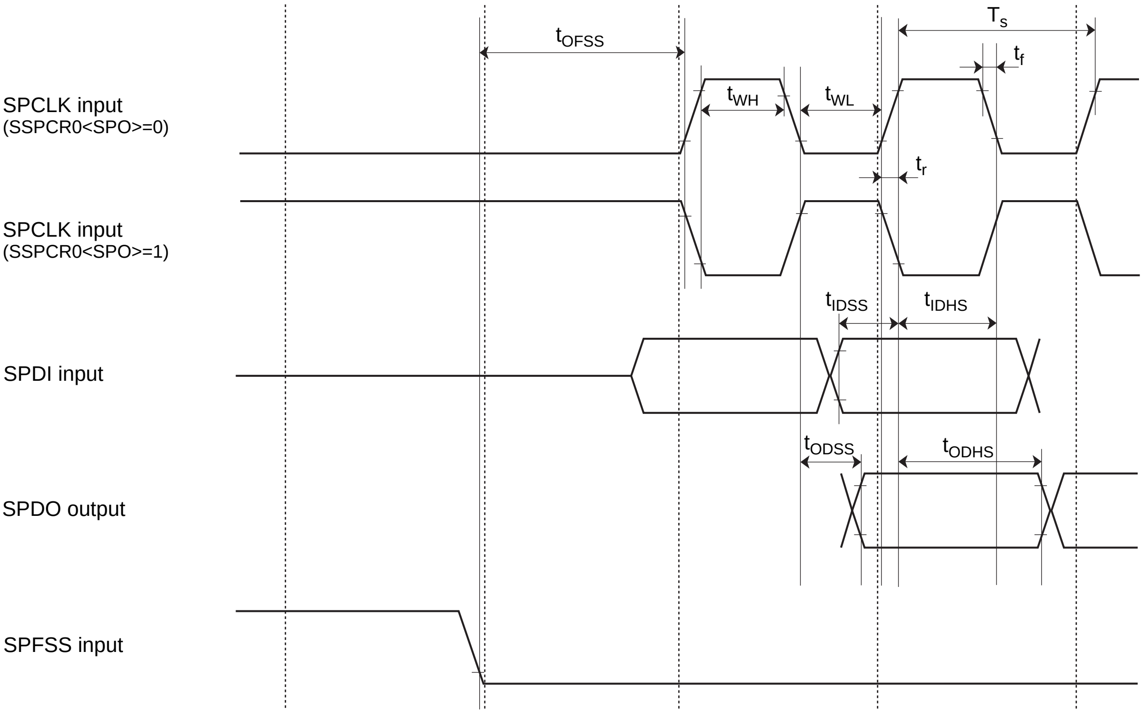
(2) Master SSPCR0<SPH>="1" (Data is latched on the second edge)



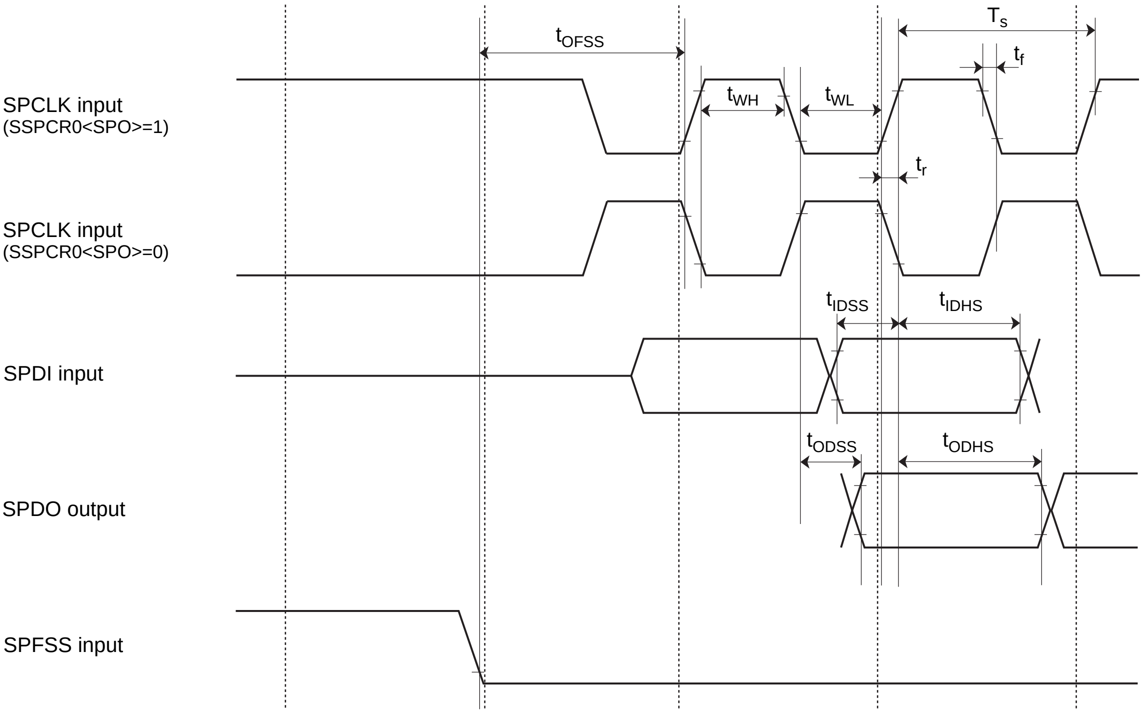
26.6.4.3 SSP SPI mode (Slave)

- fsys ≥ 12 × SPCLK (Maximum)
- fsys ≥ 65024 × SPCLK (Minimum)

(3) Slave SSPCR0<SPH>="0" (Data is latched on the first edge.)



(4) Slave SSPCR0<SPH>="1" (Data is latched on the second edge.)



26.6.5 Event Counter

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Clock low-level pulse width	t_{VCKL}	2x + 100	−	150	−	ns
Clock high-level pulse width	t_{VCKH}	2x + 100	−	150	−	ns

26.6.6 Capture

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low-level pulse width	t_{CPL}	2x + 100	−	150	−	ns
High-level pulse width	t_{CPH}	2x + 100	−	150	−	ns

26.6.7 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP release interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
INT0 to F low-level pulse width	t_{INTAL}	x + 100	−	125	−	ns
INT0 to F high-level pulse width	t_{INTAH}	x + 100	−	125	−	ns

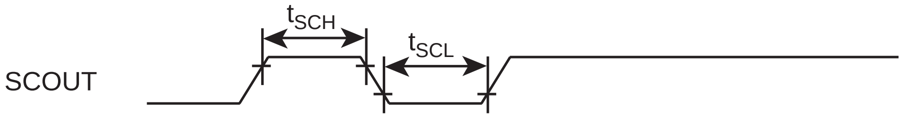
2. STOP release interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
INT0 to F low-level pulse width	t_{INTBL}	100	−	100	−	ns
INT0 to F high-level pulse width	t_{INTBH}	100	−	100	−	ns

26.6.8 SCOUT pin AC Characteristics

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
High-level pulse width	t_{SCH}	0.5T − 5	−	7.5	−	ns
Low-level pulse width	t_{SCL}	0.5T − 5	−	7.5	−	ns

Note: In the above table, the letter T represents the cycle time of the SCOUT output clock.



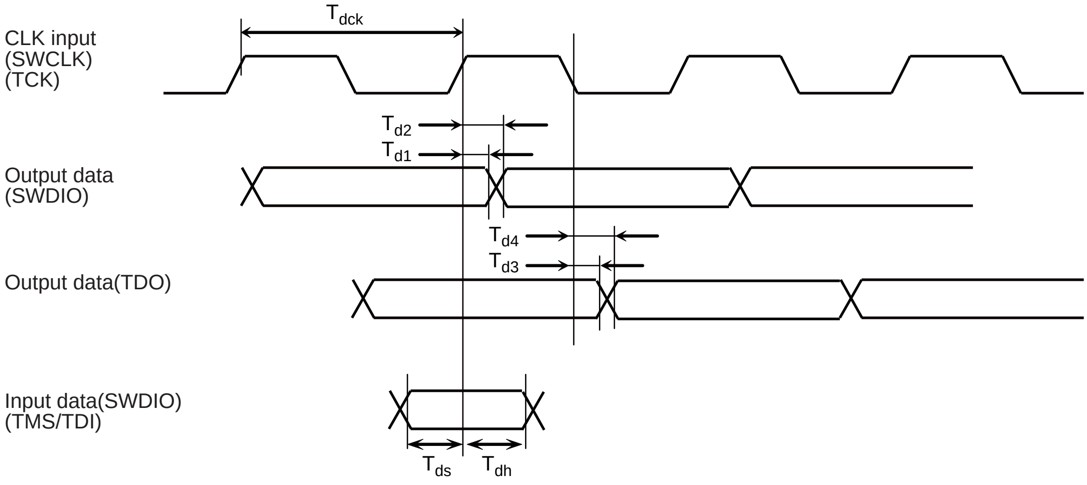
26.6.9 Debug Communication

26.6.9.1 SWD Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK rise → Output data hold	T_{d1}	4	–	
CLK rise → Output data valid	T_{d2}	–	37	
Input data valid ← CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	

26.6.9.2 JTAG Interface

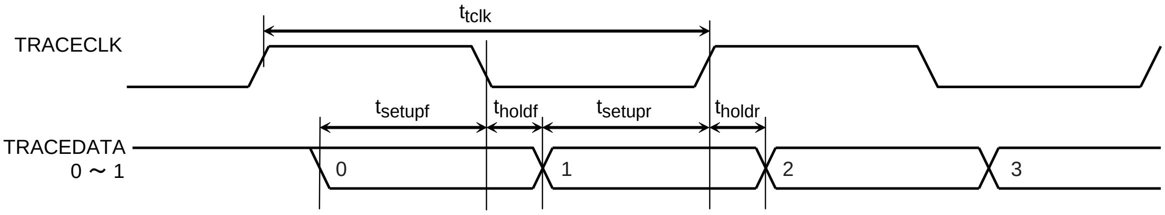
Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	–	ns
CLK fall → Output data hold	T_{d3}	4	–	
CLK fall → Output data valid	T_{d4}	–	37	
Input data valid ← CLK rise	T_{ds}	20	–	
CLK rise → Input data hold	T_{dh}	15	–	



26.6.10 ETM Trace

- Output levels: High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Load capacitance: TRACECLK = 25pF, TRACEDATA = 20pF

Parameter	Symbol	Min	Max	Unit
TRACECLK cycle	t_{tclk}	50	−	ns
TRACEDATA valid ← TRACECLK rise	t_{setupr}	2	−	
TRACECLK rise → TRACEDATA hold	t_{holdr}	1	−	
TRACEDATA valid ← TRACECLK fall	t_{setupf}	2	−	
TRACECLK fall→ TRACEDATA hold	t_{holdf}	1	−	



26.6.11 On-chip Oscillator Characteristic

Parameter	Symbol	Condition	Min	Typ.(Note 1)	Max	Unit
Oscillation frequency	IHOSC	Ta = 25°C	−	9.0	−	MHz
Oscillation accuracy	−	Ta = -40 to 85°C	-15	−	+15	%

Note 1: Ta = 25 °C, DVDD5 = RVDD5 = AVDD5 = 5 V, unless otherwise noted.

Note: Do not use an on-chip oscillator as a system clock (fsys) when high-accuracy oscillation frequency is required.

26.6.12 Flash Characteristic

Parameter	Condition	Min	Typ.	Max	Unit
Guaranteed number of Flash memory programming	Ta = 0 to 70°C	−	−	100	times

26.7 Recommended Oscillation Circuit

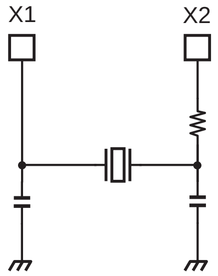


Figure 26-1 High-frequency oscillation connection

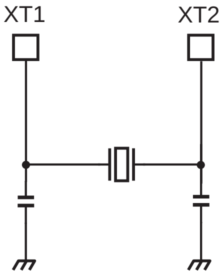


Figure 26-2 Low-frequency oscillation connection

Note: To obtain a stable oscillation, load capacity and the position of the oscillator must be configured properly. Since these factors are strongly affected by substrate patterns, please evaluate oscillation stability using the substrate you use.

The TX03 has been evaluated by the oscillator vendor below. Please refer this information when selecting external parts

26.7.1 Ceramic Oscillator

This product has been evaluated by the ceramic oscillator of Murata Manufacturing Co., Ltd.

Please refer to the Murata Website for details.

26.7.2 Crystal Oscillator

This product has been evaluated by the crystal oscillator of KYOCERA Corporation.

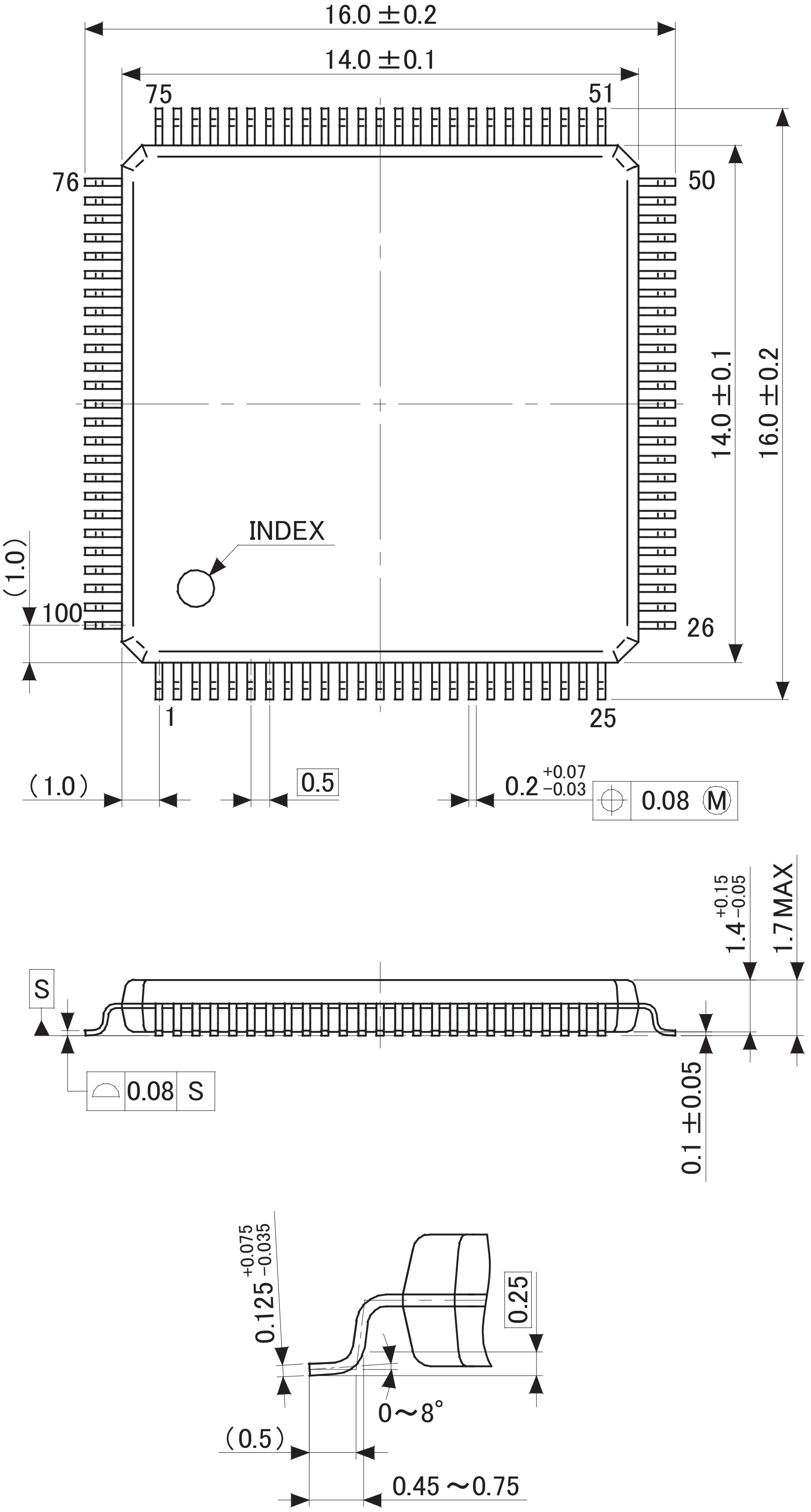
Please refer to the KYOCERA Website for details.

27. Package Dimensions

LQFP100-P-1414-0.50H

Dimensions

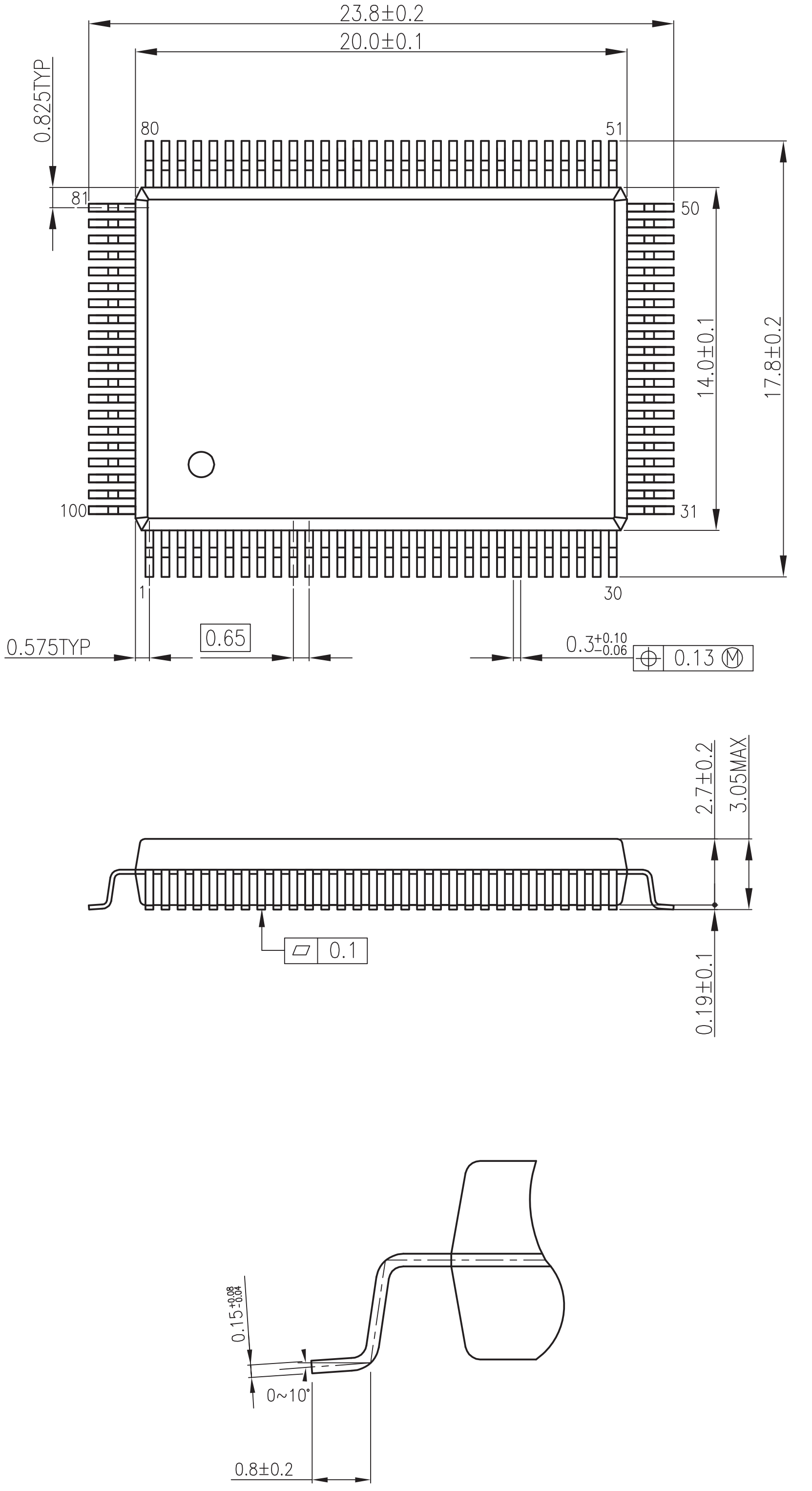
Unit: mm



QFP100-P-1420-0.65Q

Dimensions

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.**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

https://toshiba.semicon-storage.com/