TOSHIBA

32-bit RISC Microcontroller
TX03 Series

TMPM372FWUG
TMPM373FWDUG
TMPM374FWUG

TOSHIBA CORPORATION

Semiconductor & Storage Products Company

Revision History

Rev	Date	Description
1	2011/8/1	First Release

ARM

Comparison table of TMPM370 group

	TMPM370	TMPM372	TMPM373	TMPM374
CPU core	Cortex-M3	Cortex-M3	Cortex-M3	Cortex-M3
Maximum Operating Frequency	80MHz	80MHz	80MHz	80MHz
ROM (Flash)	256Kbyte	128Kbyte	128Kbyte	128Kbyte
RAM	10Kbyte	6Kbyte	6Kbyte	6Kbyte
PMD	2ch	1ch	1ch	1ch
ENC	CH0 & CH1	CH1	CH1	CH1
Amp/Comparator	Available	N/A	N/A	N/A
Internal oscillator	N/A	Available	Available	Available
12bit ADC	ADC A & ADC B 22ch	ADC B 11ch (AINB2 to AINB12)	ADC B 7ch (AINB2, AINB3, AINB8 to AINB12)	ADC B 6ch (AINB2, AINB3, AINB9 to AINB12)
Vector Engine	Available	Available	Available	Available
16bit Timer/Counter	8ch TMBR0 to 7 IN: 0 to 7 OUT: 0 to 7	8ch TMBR0 to 7 IN: 0,1,2,3,4,6,7 OUT: 0,1,2,3,4,6,7	8ch TMBR0 to 7 IN: 1,2,3,6,7 OUT: 3,6,7	8ch TMBR0 to 7 IN: 2,3,6,7 OUT: 3,6
UART/SIO	4ch CH0 to 3	4ch CH0 to 3	3ch CH0,1,3	3ch CH0,1,3
Interrupt	Internal: 62 External: 16	Internal: 49 External: 10	Internal: 45 External:8	Internal: 44 External:7
I/O	76 pins	53 pins	37 pins	33 pins
OFD	Available	Available	Available	Available
POR	Available	Available	Available	Available
VLTD	Available	Available	Available	Available
Debug Interface	JTAG,SWD/V,TRACE	JTAG,SWD/V	JTAG(non TRST),SWD/V	JTAG(non TRST),SWD/V
Operating Voltage	4.5V to 5.5V	4.5V to 5.5V	4.5V to 5.5V	4.5V to 5.5V
Operating Temperature	-40 to 85°C	-40 to 85°C(1 to 80MHz) -40 to 105°C(1 to 32MHz)	-40 to 85°C(1 to 80MHz) -40 to 105°C(1 to 32MHz)	-40 to 85°C(1 to 80MHz) -40 to 105°C(1 to 32MHz)
Package	LQFP100(14X14) QFP100(14X20)	LQFP64(10X10)	LQFP48(7X7)	LQFP44(10X10)

32-bit RISC microcontroller TX03 series
TMPM372 / 373 / 374

TX03 series is a 32-bit RISC microcontroller with an ARM®Cortex^{TM}-M3 microcontroller core.

Product No.	On Chip Flash ROM	RAM	Package
TMPM372FWUG	128Kbyte	6Kbyte	LQFP64-P-1010-0.50E
TMPM373FWDUG	128Kbyte	6Kbyte	LQFP48-P-0707-0.50C
TMPM374FWUG	128Kbyte	6Kbyte	LQFP44-P-1010-0.80A

1.1   Features

(1) ARM Cortex-M3 microcontroller core
- 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 and code size
    - Auto-switching between 32-bit instruction and 16-bit instruction is executed by compiler.
- Both high performance and low power consumption have been achieved

    High performance
    - A 32-bit multiplication (32x32=32bit) can be executed with one clock
    - Division takes between 2 and 12 cycles depending on dividend and devisor

    Low power consumption
    - Optimized design using a low power consumption library
    - Standby function that stops the operation of the microcontroller core
- High-speed interrupt response suitable for real-time control
    - An interruptible long instruction
    - Stack push automatically handled by hardware
- Interrupt source
    - TMPM372 (Internal: 49 / External: 10)
    - TMPM373 (Internal: 45 / External: 8)
    - TMPM374 (Internal: 44 / External:7)
    - The order of precedence of internal sources except the watchdog timer interrupt can be set over 7 levels
- Input / output ports
    - TMPM372 (53 pins)
    - TMPM373 (37 pins)
    - TMPM374 (33 pins)
- Watchdog timer (WDT): 1channel
- Power On reset function (POR)
- Voltage detect function (VLTD)
- Oscillation frequency detect function (OFD)
- Vector engine (VE): 1 unit
    - Calculating circuit for motor control
- Programmable motor driver (PMD): 1channel
    - 3 phase complementary PWM generator
    - Synchronous A/D convert start trigger generator

- Emergency protective function (EMG pin)
- Encoder input circuit (ENC): 1 channel
    - Correspond to incremental encoder (AB/ABZ)
    - Rotation direction detection
    - Counter for absolute position detection
    - Comparator for position detection
    - Noise filter
    - 3 phase sensor input
- 16-bit timer (TMRB): 8 channel

    16-bit interval timer mode, 16-bit event counter mode, 16-bit PPG output, Input capture function

    - TMPM372: Input 7 pins / Output 7 pins
    - TMPM373: Input 5 pins / Output 3 pins
    - TMPM374: Input 4 pins / Output 2 pins
- General-purpose serial interface (SIO)

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

    - TMPM372: 4 channel
    - TMPM373: 3 channel
    - TMPM374: 3 channel
- 12-bit A/D converter (ADC): 1unit
    - Start by the internal trigger: TMRB interrupt / PMD trigger
    - Single/repeat mode
    - AD conversion monitoring function
    - Conversion speed $2\mu s$ (@ADC conversion clock = 40 MHz)
- Standby mode
    - Standby mode: IDLE, STOP
- Clock generator (CG)
    - On-chip PLL (8 times)
    - Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8 or 1/16
- Endian
    - Little endian
- Internal high-speed oscillation circuit
- Maximum operating frequency
    - 80 MHz
- Operating voltage range
    - 4.5V to 5.5V (with on-chip regulator)
- Temperature range
    - -40 to $85°$C (except during Flash writing / 1 to 80 MHz)
    - -40 to 105°C (except during Flash writing / 1 to 32 MHz)
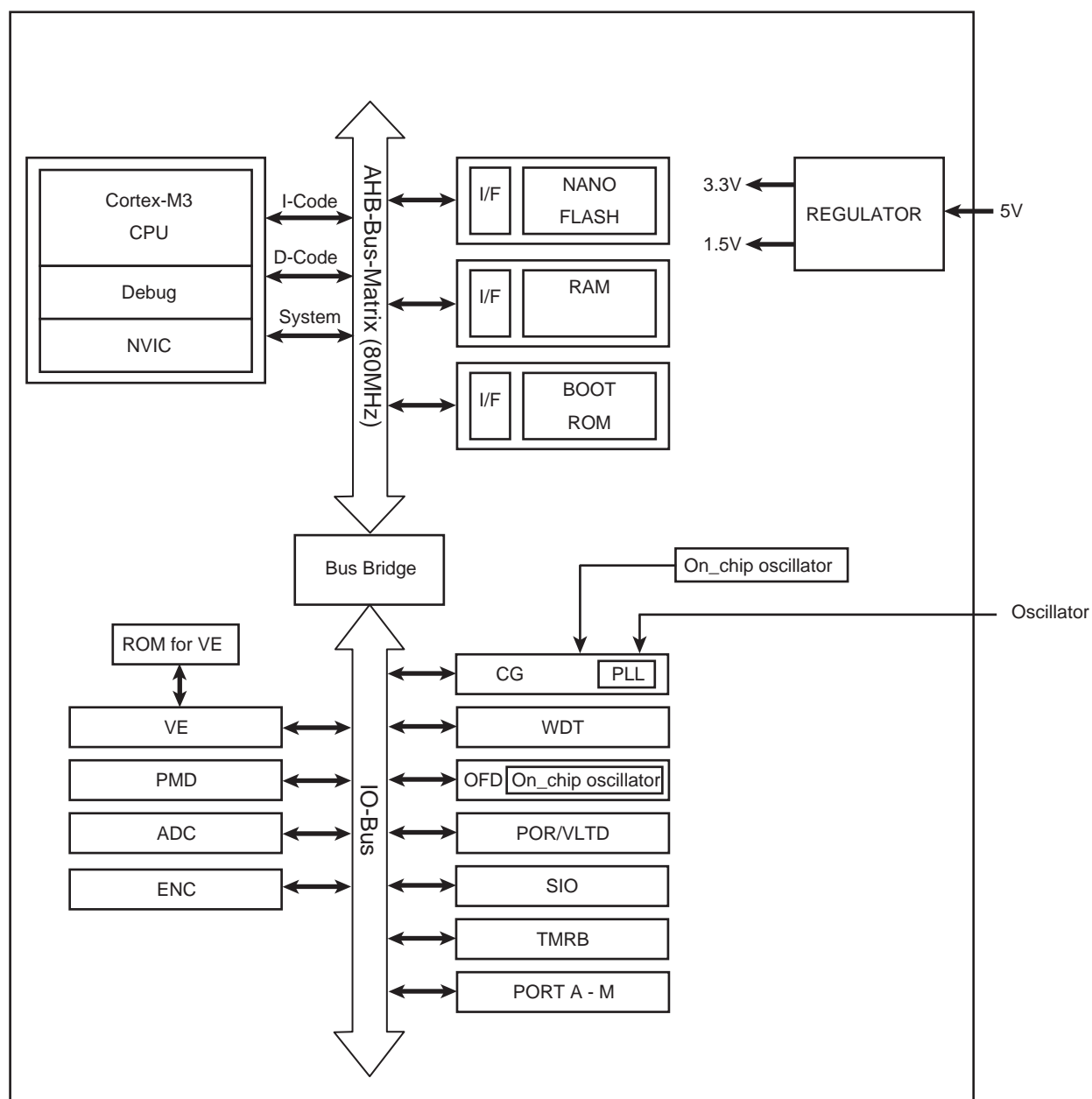   - 0 to 70 °C (during Flash writing / erasing)

1.2　Block Diagram

Figure 1-1　TMPM372 / 373 / 374 block diagram

2. Pin Layout and Pin Functions

2.1 Pin Layout (Top view)

This chapter shows a pin layout of TMPM372 / 373 / 374.
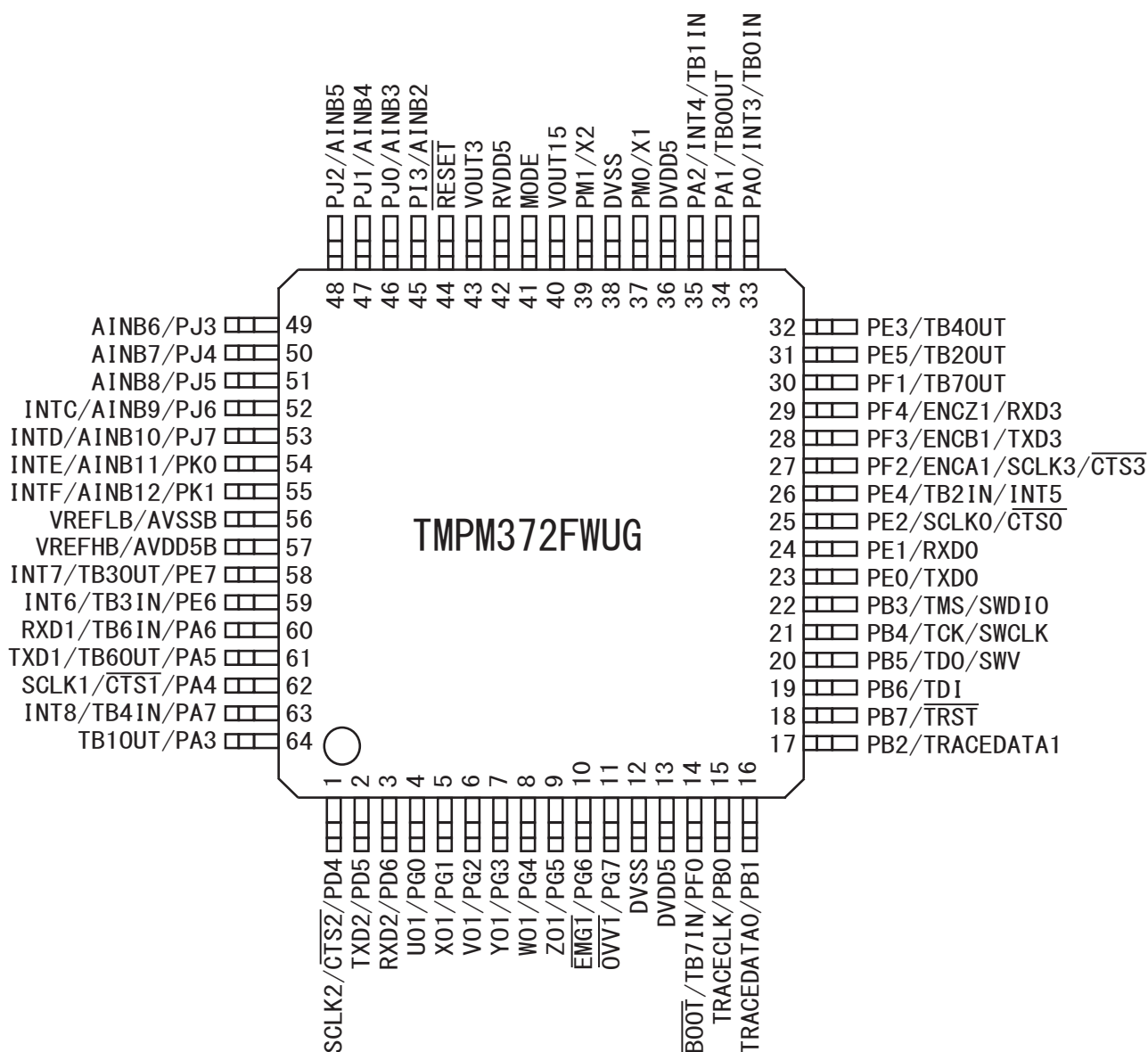
LQFP64−P−1010−0.50E
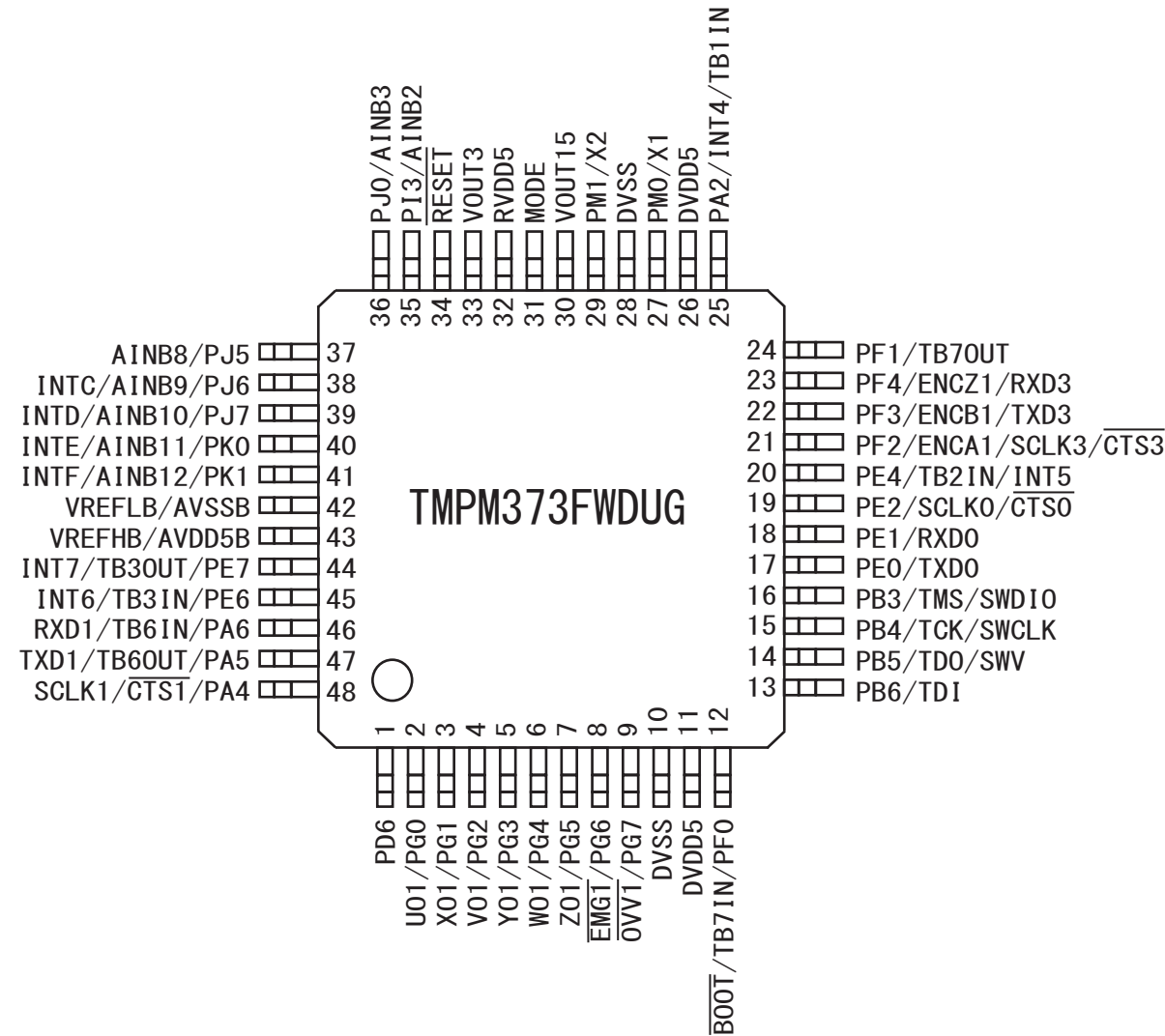

Figure 2-1 TMPM372FWUG Pin Layout

LQFP48-P-0707-0.50C

Figure 2-2 TMPM373FWDUG Pin Layout
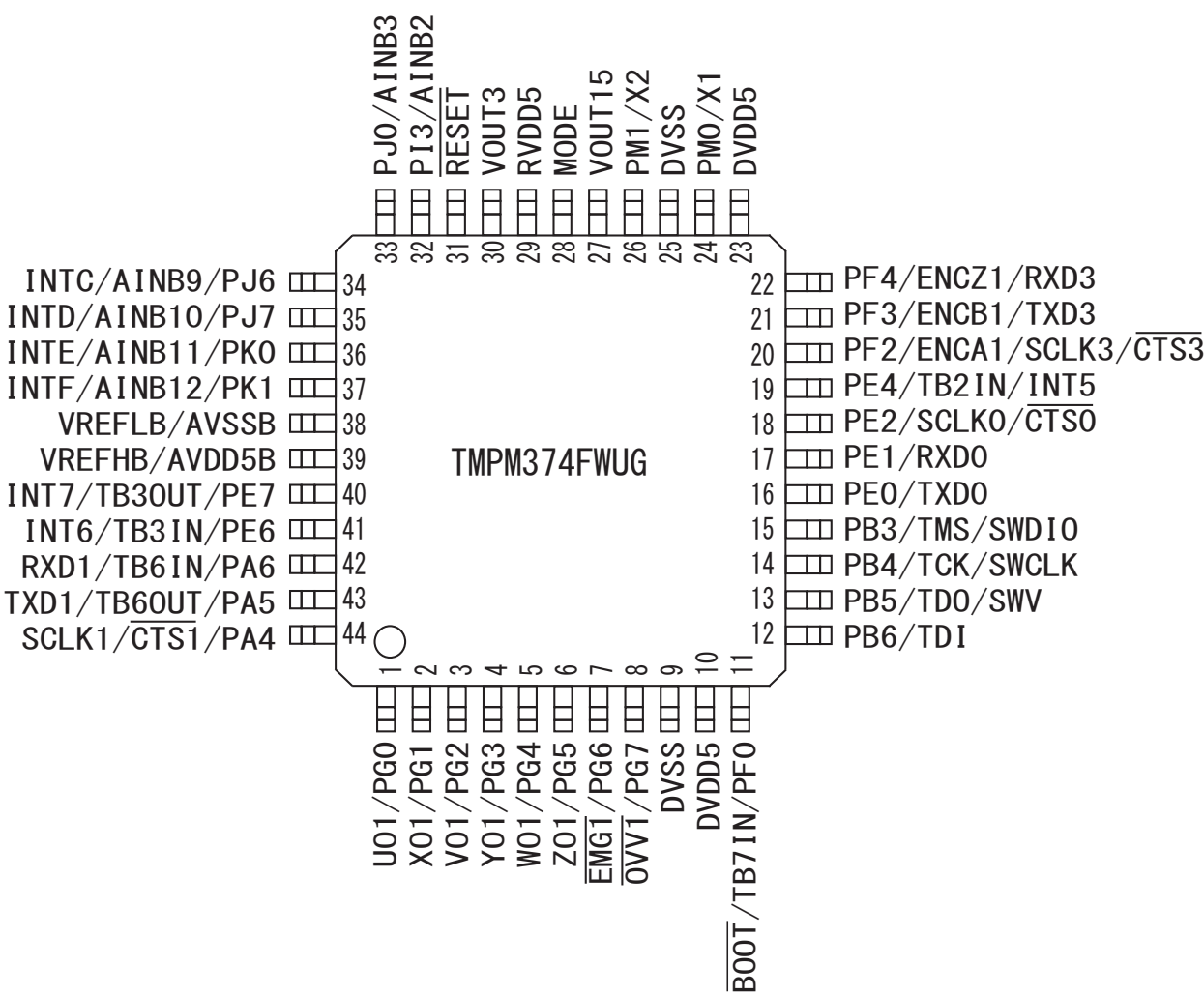
LQFP44-P-1010-0.80A

Figure 2-3 TMPM374FWUG Pin Layout

2.2   Pin function

Table 2-1 shows the pin function of TMPM372 / 373 / 374. Table 2-2 shows the operating voltage of each pin, and Table 2-3 shows the voltage range of every pin.

Table 2-1 Pin functions

Pin name	Existence of pin (O:Yes, -:No)			Output during Reset	SCHMITT (O:Yes)	Open Drain mode	High Speed
	TMPM372	TMPM373	TMPM374				
PA0 TB0IN INT3	O	-	-	Hi-Z	O	O	
PA1 TB0OUT	O	-	-	Hi-Z	O	O	
PA2 TB1IN INT4	O	O	-	Hi-Z	O	O	
PA3 TB1OUT	O	-	-	Hi-Z	O	O	
PA4 SCLK1 $\overline{CTS1}$	O	O	O	Hi-Z	O	O	
PA5 TXD1 TB6OUT	O	O	O	Hi-Z	O	O	
PA6 RXD1 TB6IN	O	O	O	Hi-Z	O	O	
PA7 TB4IN INT8	O	-	-	Hi-Z	O	O	
PB0 TRACECLK	O	-	-	Hi-Z	O	O	O
PB1 TRACEDATA0	O	-	-	Hi-Z	O	O	O
PB2 TRACEDATA1	O	-	-	Hi-Z	O	O	O
PB3 TMS SWDIO	O	O	O	Pull UP	O	O	
PB4 TCK SWCLK	O	O	O	Pull DOWN	O	O	
PB5 TDO SWV	O	O	O	Hi-Z	O	O	
PB6 TDI	O	O	O	Pull UP	O	O	
PB7 $\overline{TRST}$	O	-	-	PUll UP	O	O	
PD4 SCLK2 $\overline{CTS2}$	O	-	-	Hi-Z	O	O	

Table 2-2 Pin functions

Pin name	Existence of pin (O:Yes, -:No)			Output during Reset	SCHMITT (O:Yes)	Open Drain mode	High Speed
	TMPM372	TMPM373	TMPM374				
PD5 TXD2	O	-	-	Hi-Z	O	O	
PD6 RXD2	O	O	-	Hi-Z	O	O	
PE0 TXD0	O	O	O	Hi-Z	O	O	
PE1 RXD0	O	O	O	Hi-Z	O	O	
PE2 SCLK0 $\overline{CTS0}$	O	O	O	Hi-Z	O	O	
PE3 TB4OUT	O	-	-	Hi-Z	O	O	
PE4 TB2IN INT5	O	O	O	Hi-Z	O	O	
PE5 TB2OUT	O	-	-	Hi-Z	O	O	
PE6 TB3IN INT6	O	O	O	Hi-Z	O	O	
PE7 TB3OUT INT7	O	O	O	Hi-Z	O	O	
PF0 TB7IN $\overline{BOOT}$	O	O	O	Pull UP	O	O	
PF1 TB7OUT	O	O	-	Hi-Z	O	O	
PF2 SCLK3 $\overline{CTS3}$ ENCA1	O	O	O	Hi-Z	O	O	
PF3 TXD3 ENCB1	O	O	O	Hi-Z	O	O	
PF4 RXD3 ENCZ1	O	O	O	Hi-Z	O	O	
PG0 UO1	O	O	O	Hi-Z	O	O	
PG1 XO1	O	O	O	Hi-Z	O	O	
PG2 VO1	O	O	O	Hi-Z	O	O	
PG3 YO1	O	O	O	Hi-Z	O	O	
PG4 WO1	O	O	O	Hi-Z	O	O	
PG5 ZO1	O	O	O	Hi-Z	O	O	
PG6 $\overline{EMG1}$	O	O	O	Hi-Z	O	O	

Table 2-2 Pin functions

Pin name	Existence of pin (O:Yes, -:No)			Output during Reset	SCHMITT (O:Yes)	Open Drain mode	High Speed
	TMPM372	TMPM373	TMPM374				
PG7 OVV1	O	O	O	Hi-Z	O	O	
PI3 AINB2	O	O	O	Hi-Z	O	O	
PJ0 AINB3	O	O	O	Hi-Z	O	O	
PJ1 AINB4	O	-	-	Hi-Z	O	O	
PJ2 AINB5	O	-	-	Hi-Z	O	O	
PJ3 AINB6	O	-	-	Hi-Z	O	O	
PJ4 AINB7	O	-	-	Hi-Z	O	O	
PJ5 AINB8	O	O	-	Hi-Z	O	O	
PJ6 AINB9 INTC	O	O	O	Hi-Z	O	O	
PJ7 AINB10 INTD	O	O	O	Hi-Z	O	O	
PK0 AINB11 INTE	O	O	O	Hi-Z	O	O	
PK1 AINB12 INTF	O	O	O	Hi-Z	O	O	
PM0 X1	O	O	O	Hi-Z	O	O	
PM1 X2	O	O	O	Hi-Z	O	O	
MODE	O	O	O	Hi-Z	O		
RESET	O	O	O	Pull UP	O		
DVDD5	O	O	O				
RVDD5	O	O	O				
DVSS	O	O	O				
AVSSB VREFLB	O	O	O				
AVDD5B VREFHB	O	O	O				
VOUT3	O	O	O				
VOUT15	O	O	O				

Table 2-3 Operating voltage of each Pin

Pin name	Operating voltage	
X1, X2	Internal 1.5V/DVSS	Do not be driven by external circuit.
$\overline{RESET}$	DVDD5/DVSS	
MODE	DVDD5/DVSS	Must be connected to GND
I/O	DVDD5/DVSS	
AIN	AVDD5B/AVSS5B	

Table 2-4 Voltage range of each pin

Pin name	Voltage range	
DVDD5		For I/O ports
RVDD5	4.5 to 5.5V	For internal circuit
AVDD5B		For ADC
VOUT15	1.35 to 1.65V	VOUT15 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit. (Note)
VOUT3	2.7 to 3.6V	VOUT3 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit. (Note)
DVSS	GND	
AVSSB		

Note 1:VOUT15 and VOUT3 must be connected with the same value of capacitors.

3. Processor Core

3.1 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.

The following table shows the revision of the processor core in the TMPM372 / 373 / 374. For further information on each revision, see the documents issued by ARM Limited.

Product Name	Core Revision
TMPM372 / 373 / 374	r2p0

3.2 Optional block

The Cortex-M3 core has the optional blocks. The optional blocks of the TMPM372 / 373 / 374 are below;

Optional block	Implementation
FPB	O
DWT	O
ITM	O
MPU	-
ETM™	O
AHB-AP	O
AHB trace macro-cell interface	O
TPIU	O
WIC	-

O : Implement, -:Not implement

3.3 Event

TMPM372 / 373 / 374 doesn't support event input/output. Do not use SEV instruction and WFE instruction.

3.4 SLEEPDEEP

TMPM372 / 373 / 374 doesn't support SLEEPDEEP. Do not set SLEEPDEEP bit.

Event signal is not supported also. Do not use WFE instruction.

3.5 Exclusive access

TMPM372 / 373 / 374 doesn't support EXCLUSIVE ACCESS.

3.6 Reset operation

3.6.1 Initial state

The internal circuits, register settings and pin status are undefined right after the power-on. The state continues until the $\overline{\text{RESET}}$ pin receives "Low" level signal after all the power supply voltage is applied.

3.6.2 Reset operation

TMPM372 / 373 / 374 has Power-on reset circuit, power-on reset signal is generated when power supply is turned on.

When reset from external $\overline{\text{RESET}}$ pin, as the precondition, ensure that a high-speed oscillator provides stable oscillation while power supply voltage is in the operating range.

To reset the TMPM372 / 373 / 374, input reset signal to $\overline{\text{RESET}}$ pin at "Low" level for minimum duration of 12 system clocks.

3.6.3 After Reset

When the reset is released, the system control register and the internal I/O register of the Cortex-M3 processor core are initialized. Note that the PLL multiplication circuit stops after releasing the reset. Therefore, set CGOSCCR register and CGPLLSEL register to use PLL multiplication circuit.

After the reset exception handling is executed, the program branches off to the interrupt service routine. The address with which the interrupt service routine starts is stored in 0x0000_0004.

Note 1:It is possible to turn power on after $\overline{\text{RESET}}$ pin is set to "Low".

Note 2:The rest operation may after the internal RAM state.

4. Debug Interface

4.1 Specification Overview

The TMPM372 / 373 / 374 contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the Debug interface and the Embedded Trace Macrocell^{TM} (ETM) unit for trace output. Trace data is output to the dedicated pins (TRACEDATA[0] to [1], SWV) via the on-chip Trace Port Interface Unit (TPIU).

Note :TMPM373 / 374 don't have TRACEDATA[0], [1] and TRACECLK pins.

4.2 Features of SWJ-DP

SWJ-DP supports the two-pin Serial Wire Debug Port (SWDCK, SWDIO) and the JTAG Debug Port (TDI, TDO, TMS, TCK, $\overline{TRST}$).

Note :TMPM373 / 374 don't have $\overline{TRST}$ pin

4.3 Features of ETM

ETM supports two data signal pins (TRACEDATA[0] to [1]), one clock signal pin (TRACECLK) and trace output from SWV.

Note :TMPM373 / 374 don't have TRACEDATA[0], [1] and TRACECLK pins.

4.4 Pin Functions

The debug interface pins can also be used as general-purpose ports. The PB3 and PB4 are shared between the JTAG debug port function and the serial wire debug port function. The PB5 is shared between the JTAG debug port function and the SWV trace output function.

Table 4-1 SWJ-DP, ETM function

SWJ-DP Pin name	Name of port	Existence of pin (O:Yes, -:No)			JTAG debug function		SW debug	
		TMPM372	TMPM373	TMPM374	I/O	Description	I/O	Description
TMS/SWDIO	PB3	O	O	O	Input	JTAG Test Mode Selection	I/O	Serial Wire Data Input/Output
TCK/SWCLK	PB4	O	O	O	Input	JTAG Test Check	Input	Serial Wire Clock
TDO/SWV	PB5	O	O	O	Output	JTAG Test Data Output	(Input) (Note1)	(Serial Wire Viewer Output)
TDI	PB6	O	O	O	Input	JTAG Test Data Input	-	-
$\overline{TRST}$	PB7	O	-	-	Input	JTAG Test nRESET	-	-
TRACECLK	PB0	O	-	-	Output	TRACE Clock Output		
TRACEDATA0	PB1	O	-	-	Output	TRACE DATA Output0		
TRACEDATA1	PB2	O	-	-	Output	TRACEDATA Output1		

Note :In case of enabling SWV function

After reset, the PB3, PB4, PB5, PB6 and PB7 are configured as debug port function pins. The functions of other debug interface pins need to be programmed as required. Debug interface pins can use general purpose port that is not use debug interface.

Table 4-2 below summarizes the debug interface pin functions and related port settings after reset.

Table 4-2 Debug interface pins and port setting after reset

Initial Setting	Port (Bit name)	Debug Function	Existence of pin (O:Yes, -:No)			Port Setting After Reset (-;No register)					
			TMPM372	TMPM373	TMPM374	Function (PBFR)	Input (PBIE)	Output (PBCR)	Open Drain (PBOD)	Pull-up (PBPUP)	Pull-down (PBPDN)
PORT	PB0	TRACECLK	O	-	-	0	0	0	0	0	0
PORT	PB1	TRACEDATA0	O	-	-	0	0	0	0	0	0
PORT	PB2	TRACEDATA1	O	-	-	0	0	0	0	0	0
DEBUG	PB3	TMS/SWDIO	O	O	O	1	1	1	0	1	0
DEBUG	PB4	TCK/SWCLK	O	O	O	1	1	0	0	0	1
DEBUG	PB5	TDO/SWV	O	O	O	1	0	1	0	0	0
DEBUG	PB6	TDI	O	O	O	1	1	0	0	1	0
DEBUG	PB7	$\overline{TRST}$	O	-	-	1	1	0	0	1	0

When using a low power consumption mode, take note of the following points.

Note 1:If PB3 and PB5 are configured as debug function pins, output continues to be enabled even in STOP mode regardless of the setting of the CGSTBYCR<DRVE>.

Note 2:If PB4 is configured as a debug function pin, it prevents a low power consumption mode from being fully effective. Configure PB4 to function as a general-purpose port if the debug function is not used.

4.5 Connection with a Debug Tool

4.5.1 How to connect

For how to connect a debug tool, refer to the method recommended by each manufacture.Debug interface pins have pull-up or pull-down register. When connect with pull-up or pull-down riggers, be sure their settings.

4.5.2 When use general purpose port

When debugging, do not change setting debug interface to general purpose port by program. Then, MCU will be unable to control signals received from the debugging tools and can not continue debugging. According to the usage of the debug interface pins, be sure their setting.

Table 4-3 Debug Interface

Usage	Using Debug Interface (O:Enable, -:Disable)							
	$\overline{\text{TRST}}$	TDI	TDO/ SWV	TCK/ SWCLK	TMS/ SWDIO	TRACE DATA1	TRACE DATA0	TRACE CLK
JTAG+SW (After RESET)	O	O	O	O	O	-	-	-
JTAG+SW (non $\overline{\text{TRST}}$)	-	O	O	O	O	-	-	-
JTAG+TRACE	O	O	O	O	O	O	O	O
SW	-	-	-	O	O	-	-	-
SW+SWV	-	-	O	O	O	-	-	-
Disable Debug function	-	-	-	-	-	-	-	-

4.6 Peripherals operation during HALT mode

When Break during debugging, Cortex-M3 CPU core going into HALT mode. Watch dog timer (WDT) is stopped counting automatically. And 16bit timer/counter can specify the status (continue operating or stop) in HALT mode. Other peripherals are continue operating.

5. Memory Map

5.1 Memory Map

The memory maps for the TMPM372 / 373 / 374 are based on the ARM Cortex-M3 processor core memory map.The internal ROM, internal RAM and internal I/O regions of the TMPM372 / 373 / 374 are mapped to the code, SRAM and peripheral regions of the Cortex-M3 respectively. The SRAM and internal I/O regions are all included in the bit-band region.

The CPU register region is the processor core's internal register region. For more information on each region, see the "Cortex-M3 Technical Reference Manual".

Note that regions indicated as "Fault" is accessed, memory fault is generated if memory fault is enable, or hard fault is generated if memory fault is disable. Do not access the vendor-specific regions.
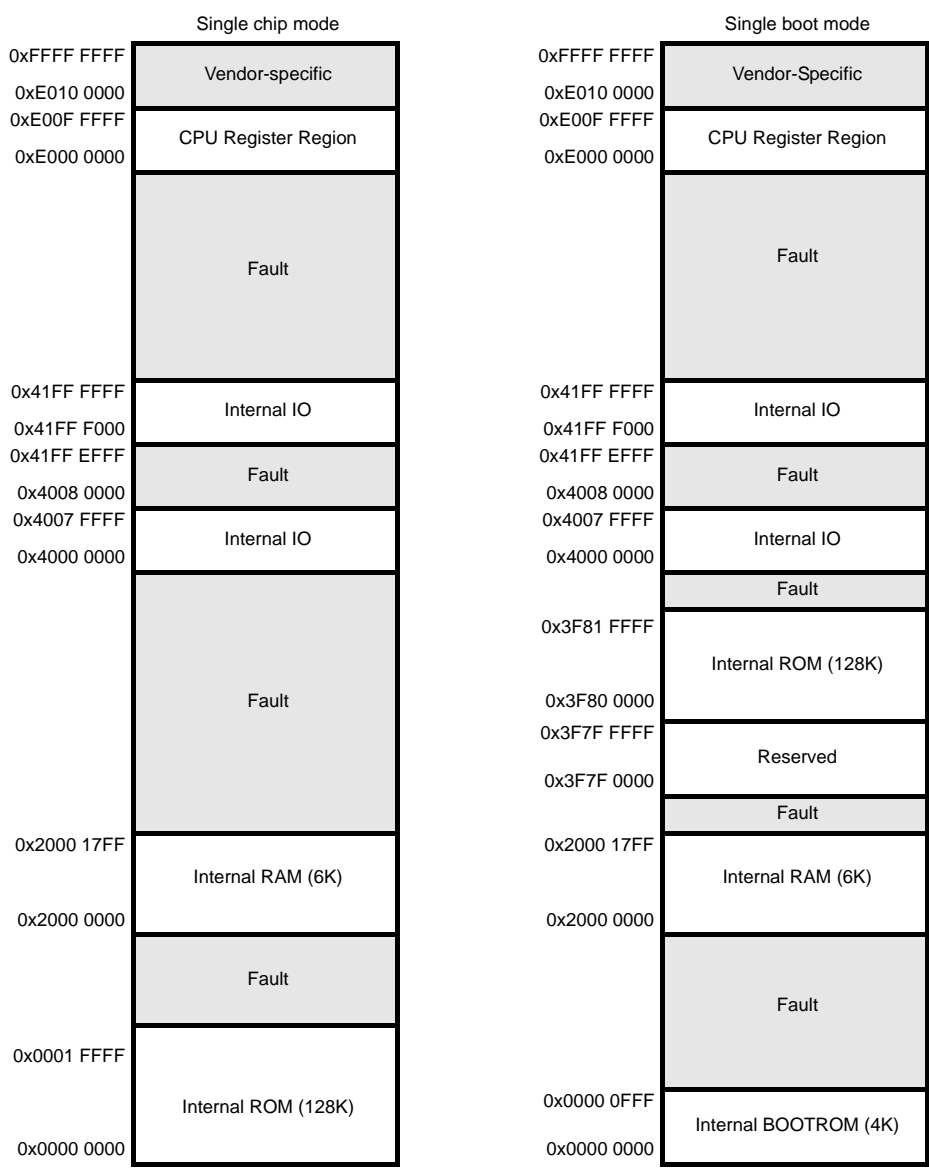
Single chip mode		Single boot mode	
0xFFFF FFFF	Vendor-specific	0xFFFF FFFF	Vendor-Specific
0xE010 0000		0xE010 0000	
0xE00F FFFF	CPU Register Region	0xE00F FFFF	CPU Register Region
0xE000 0000		0xE000 0000	
	Fault		Fault
0x41FF FFFF	Internal IO	0x41FF FFFF	Internal IO
0x41FF F000		0x41FF F000	
0x41FF EFFF	Fault	0x41FF EFFF	Fault
0x4008 0000		0x4008 0000	
0x4007 FFFF	Internal IO	0x4007 FFFF	Internal IO
0x4000 0000		0x4000 0000	
			Fault
		0x3F81 FFFF	Internal ROM (128K)
	Fault	0x3F80 0000	
		0x3F7F FFFF	Reserved
		0x3F7F 0000	
			Fault
0x2000 17FF	Internal RAM (6K)	0x2000 17FF	Internal RAM (6K)
0x2000 0000		0x2000 0000	
	Fault		Fault
0x0001 FFFF	Internal ROM (128K)	0x0000 0FFF	Internal BOOTROM (4K)
0x0000 0000		0x0000 0000	

Figure 5-1 Memory Map

6 Clock/Mode Control

6.1. Features

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

The low power consumption mode can reduce power consumption.

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

The clock/mode control block has the following functions:

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

In addition to NORMAL mode, the TMPM372/3/4FW can operate in two types of low power mode to reduce power consumption according to its usage conditions.

6.2. Registers

6.2.1. Register List

Table 6-1 shows registers and addresses of the clock generator.

Table 6-1 Registers of Clock Generator

Register name		Address
System control register	CGSYSCR	0x4004_0200
Oscillation control register	CGOSCCR	0x4004_0204
Standby control register	CGSTBYCR	0x4004_0208
PLL selection register	CGPLLSEL	0x4004_020C
System clock selection register	CGCKSEL	0x4004_0210

6.2.2. Detailed Description of Registers

6.2.2.1. System Control Register (CGSYSCR)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	GEAR2	GEAR1	GEAR0
Read/Write			R			R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function			"0" is read.			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		
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	FPSEL	-	PRCK2	PRCK1	PRCK0
Read/Write		R		R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function		"0" is read.		Fperiph clock 0:fgear 1:fc	"0" is read.	Prescaler clock 000: fperiph 　　　 100: fperiph/16 001: fperiph/2 　 101: fperiph/32 010: fperiph/4 　 110: Reserved 011: fperiph/8 　 111: Reserved		
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
Read/Write				R			R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function				"0" is read.			Write "01"	
	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function				"0" is read.				

<Bit 2:0><GEAR 2:0>   : Specifies the high-speed clock (fc) gear.

<Bit 10:8><PRCK 2:0>   : Specifies the prescaler clock to peripheral I/O.

<Bit 12><FPSEL>   : Specifies the source clock to fperiph.

6.2.2.2. Oscillation Control Register (CGOSCCR)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
Read/Write	R	R	R	R	R/W	R/W	R	W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.				Clock source for Warm-up timer Write "0".	PLL operation 0: Stop 1: Oscillation	Status of Warm-up timer (WUP) 0: warm-up completed 1: Warm-up in operation	Operation of warm-up timer (WUP) 0: don't care 1: starting warm-up

	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	XEN1
Read/Write	R/W				R		R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Write "0".				"0" is read.		Write "0".	High-speed oscillator 1 (External) 0: Stop 1: Oscillation

	23	22	21	20	19	18	17	16
Bit symbol	WUODR5	WUODR4	WUODR3	WUODR2	WUPSEL2	HOSCON	OSCSEL	XEN2
Read/Write	R/W				R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	Bit5:2 for warm-up counter value.				Clock source for Warm-up timer (WUP) 0: Internal (OSC2) 1: External (OSC1)	Port M or external oscillator (X1/X2) 0: PORT M 1: External oscillator (X1/X2)	Selection of high-speed oscillator 0: Internal (OSC2) 1: External (OSC1)	High-speed oscillator 2 (Internal) 0: Stop 1: Oscillation

	31	30	29	28	27	26	25	24
Bit symbol	WUODR13	WUODR12	WUODR11	WUODR10	WUODR9	WUODR8	WUODR7	WUODR6
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	Bit13:6 for warm-up counter value.							

<Bit 0><WUEON>   : Enables to start the warm-up timer.
<Bit 1><WUEF>   : Enables to monitor the status of the warm-up timer.
<Bit 2><PLLON>   : Specifies operation of the PLL.
                          It stops after reset. Setting the bit is required.
<Bit 3><WUPSEL1>   : Write "0" to WUPSEL1.
<Bit 8><XEN1>   : Specifies operation of the high-speed oscillator 1 (OSC1).
<Bit 16><XEN2>   : Specifies operation of the high-speed oscillator 2 (OSC2).
<Bit 17><OSCSEL>   : Select high-speed oscillator between external oscillator (OSC1) and internal
                          oscillator (OSC2). Confirm <OSCSEL> become "1" then halt the OSC2
                          immediately after switching over to OSC1. Do not select OSC2 again after
                          switching to OSC1.
<Bit 18><HOSCON>   : Specifies Port M or X1/X2. When the external oscillator (OSC1) is used, Port M
                          registers (PMCR/PMPUP/PMPDN/PMIE) should be disabled. After reset, the
                          port M registers are disabled.
<Bit 19><WUPSEL2>   : Select source clock for warm-up timer between external oscillator (OSC1) and
                          internal oscilator (OSC2).

<Bit 31: 20><WUODR 13:2> : Warm-up cunter value.

The warm-up counter consists of 16-bit counter. The warm-up period lasts until the upper 12-bit of the counter value corresponds to the values specified in <WUODR 13:2>.

Note: When the HOSCON is set to "1", the registers for Port M (PMCR/PMPUP/PMPDN/PMIE) can not be accessed and the read data from these registers are always "0". If one of the Port M registers except PMDATA and PMOD is not equal to the initial value, the HOSCON can not be set to "1".

6.2.2.3. Standby Control Register (CGSTBYCR)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	STBY2	STBY1	STBY0
Read/Write			R			R/W	R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function			"0" is read.			Low power consumption mode 000: Reserved 001: STOP 010: Reserved 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved		
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	RXEN
Read/Write			R				R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function			"0" is read.				Write "0".	High-speed oscillator after releasing STOP mode 0: Stop 1: Oscillation
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	DRVE
Read/Write			R				R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function			"0" is read.				Write "0".	Pin status in STOP mode 0: Active 1: Inactive
	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
Read/Write				R				
After reset	0	0	0	0	0	0	0	0
Function				"0" is read.				

<Bit 2:0><STBY2:0>   : Specifies the low power consumption mode.
                          To enter the STOP mode, disable the oscillation (OSC1 or OSC2) which is
                          unused as system clock.

<Bit 8><RXEN>   : Specifies the high-speed oscillator operation after releasing the STOP mode.

<Bit 16><DRVE>   : Specifies the pin status in the STOP mode.

6.2.2.4. PLL Selection Register (CGPLLSEL)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	-	PLLSEL
Read/Write	R/W							R/W
After reset	0	0	1	1	1	1	1	0
Function	Write "0011111"							Select PLL output 0: fosc 1: fpll

	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R/W				R	R/W	R/W	R/W
After reset	1	0	1	0	0	0	0	1
Function	Write "1010"				"0" is read.	Write "001"		

	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							

	31	30	29	27	26	25	24	23
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							

<Bit 0><PLLSEL> : Specifies use or disuse of the clock multiplied by the PLL.
"fosc" is automatically set after reset. Resetting is required when using the PLL.

(Note1) **Be sure to disable OFD when enabling PLL. Especially, be careful setting PLL in the initial routine because OFD is available after OFD reset.**

Example :
OFDCR1<OFDWEN7:0>="0xF9" : **Sets control register write enable code**
OFDCR2<OFDEN7:0>="0x00" : **Disable OFD**
CGPLLSEL<PLLSEL>="1" : **Enable PLL**
OFDMNPLLON="xxxx" : **Sets lower detection frequency**
OFDMXPLLON="yyyy" : **Sets higher detection frequency**
OFDCR2<OFDEN7:0>="0xE4" : **Enable OFD**
OFDCR1<OFDWEN7:0>="0x06" : **Sets control register write disable code**

(Note2) **The maximum operating frequency is 32 MHz when using within the range from 85 to 100 ℃. Be careful not to exceed the maximum operating frequency by dividing system clock when using PLL.**

6.2.2.5. System Clock Selection Register (CGCKSEL)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	SYSCK	SYSCK FLG
Read/Write	R						R/W	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.						System clock 0:High-speed (fc) 1: Reserved	System clock status 0:High-speed (fc) 1: -
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

<Bit 0><SYSCKFLG>   : Shows the status of the system clock.

<Bit 1><SYSCK>   : Write "0" to SYSCK.

6.3. Clock Control

6.3.1. Clock System Block Diagram

Fig.6-1 shows the clock system diagram. Each clock is defined as follows.

fosc1	: Clock input from external high-speed oscillator (X1 and X2)
fosc2	: Clock input from internal high-speed oscillator
fosc	: High-speed clock specified by CGOSCCR<OSCSEL>
fpll	: Clock octupled by PLL
fc	: Clock specified by CGPLLSEL<PLLSEL> (high-speed clock)
fgear	: Clock specified by CGSYSCR<GEAR2:0>
fsys	: The same clock as fgear (system clock)
fperiph	: Clock specified by CGSYSCR<FPSEL>
ΦT0	: Clock specified by CGSYSCR<PRCK2:0> (prescaler clock)

The high-speed clock gear (fgear) and the prescaler clock ΦT0 are dividable.

- High-speed clock gear : fc, fc/2, fc/4, fc/8, fc/16

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

6.3.2. Initial Values after Reset

Reset initializes the clock configuration as follows.

High-speed oscillator 1 (External)	: OFF (stop)
High-speed oscillator 2 (Internal)	: ON (oscillating)
PLL (phase locked loop circuit)	: OFF (stop)
High-speed clock gear	: fc (no frequency dividing)

Reset causes all the clock configurations to be the same as fosc2.

fc = fosc2

fsys = fc (=fosc2)

fperiph = fc (=fosc2)

ΦT0 = fperiph (=fosc2)
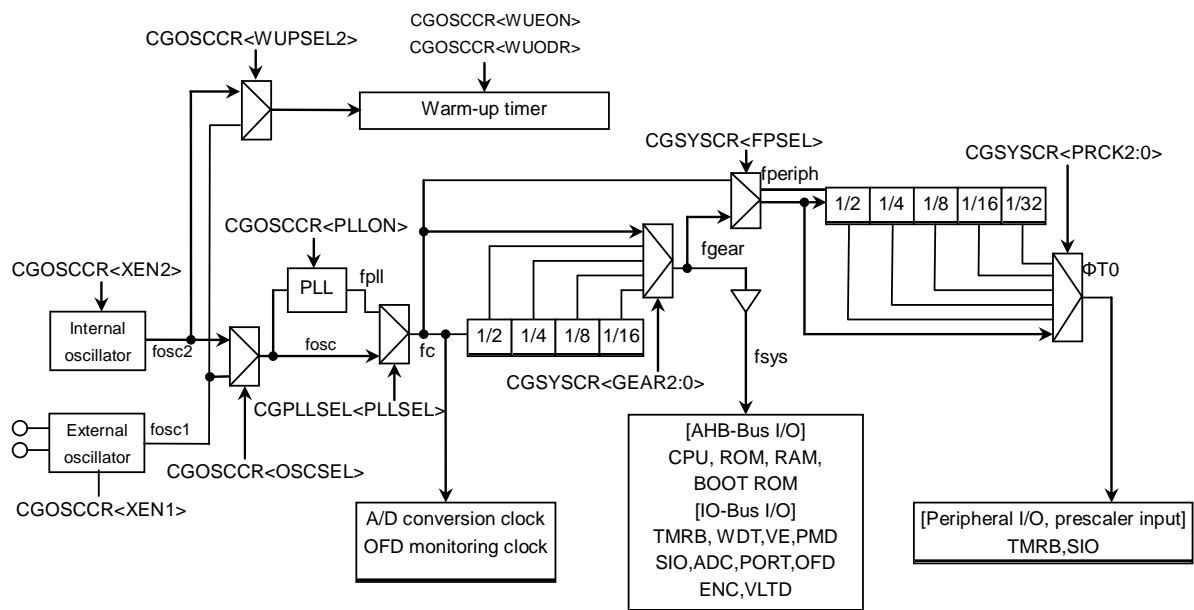
Fig.6-1 Clock Block Diagram

> **(Note)   The input clocks to selector shown with an arrow are set as default after reset.**

6.3.3. Clock Multiplication Circuit (PLL)

This circuit outputs the fpll clock that is octuple of the high-speed oscillator output clock, fosc. This lowers the oscillator input frequency while increasing the internal clock speed.

The PLL is disabled after reset is released. To enable the PLL, set "1" to the CGOSCCR<PLLON> bit. The PLL requires a certain amount of time to be stabilized, which should be secured using the warm-up function.

> **(Note)   It takes approx. 200μs for the PLL to be stabilized**

6.3.4. Warm-up Function

The warm-up function secures the stability time for the oscillator and the PLL with the warm-up timer.

The warm-up function is used when returning from STOP mode. In this case, an interrupt for returning from the low power consumption mode triggers the automatic timer count. After the specified time is reached, the system clock is output and the CPU starts operation.

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

<u>How to configure the warm-up function</u>

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

The warm-up time can be specified by setting the CGOSCCR<WUODR13:2>. The CGOSCCR<WUEON><WUEF> is used to confirm the start and completion of warm-up through software (instruction).

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

The following are the examples of the warm-up function configuration.

<Example Securing the stability time fro the PLL> (fc = fosc1)
CGOSCCR<WUPSEL1>="0"   : Write "0" to CGOSCCR<WUPSEL1>
CGOSCCR<WUPSEL2>="1"   : Specify the clock source for warm-up timer
CGOSCCR<WUPODR13:2>= "Warm-up time" / (1/fosc) / 16
                              : Specify the warm-up time
CGOSCCR<WUEON>="1"   : Start the warm-up timer (WUP)
CGOSCCR<WUEF> Read   : Wait until the state becomes "0" (warm-up is finished)

(Note)   **The warm-up counter consists of 16-bit counter. The warm-up period lasts until the upper 12-bit of the counter value corresponds to the values specified in <WUODR 13:2>.**

6.3.5. System Clock

The TMPM372/3/4 offers high-speed clock as system clock. System clock is selectable from internal oscillator or external oscillator. After reset, internal oscillator is enabled and external oscillator is disabled. The high-speed clock is dividable.

- Input frequency from X1 and X2: 8MHz to 10MHz

- Internal oscillator frequency : 10MHz

- Clock gear:1/1, 1/2, 1/4, 1/8, 1/16 (after reset: 1/1)

Table 6-2 Range of High-frequency (Unit:MHz)

Input frequency		Min. operating freq.	Max. operating freq.	After reset (PLL=OFF, CG=1/1)	Clock gear (CG) : PLL=ON					Clock gear (CG) : PLL=OFF				
					1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
OSC1	8MHz	1MHz	80MHz	8MHz	64	32	16	8	4	8	4	2	1	**
	10MHz			10MHz	80	40	20	10	5	10	5	2.5	1.25	**
OSC2	10MHz			10MHz	80	40	20	10	5	10	5	2.5	1.25	**

Note 1 : PLL=ON/OFF setting: available in CGOSCCR<PLLON>
Note 2 : Clock gear setting: available in CGSYSCR<GEAR2:0>
Note 3 : **: Reserved
Note 4 : The maximum operating frequency is 32 MHz when using within the range from 85 to 100 °C. Be careful not to exceed the maximum operating frequency by dividing system clock when using PLL.

(Note) **Switching of clock gear is executed when a value is written to the CGSYSCR<GEAR2:0> register. The actual switching takes place after a slight delay.**

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

(1) Disables port M registers (PMCR/PMPUP/PMPDN/PMIE). After reset, these registers are disabled.
(2) Specifies proper warm-up time for the external oscillator in to CGOSCCR[31:20]<WUODR13 to 2>.
(3) Sets CGOSCCR<HOSCON> to "1" to switch over from the port M to oscillator connection pins.
(4) Sets CGOSCCR<XEN1> to "1" to enable the external oscillator.
(5) Sets CGOSCCR<WUPSEL2> to "1" to specify the external oscillator clock as source clock for warm-up counter.
(6) Sets CGOSCCR<WUEON> to "1" to start warm-up and waits till the end of warm-up (CGOSCCR<WUEF> becomes "0").
(7) Sets CGOSCCR<OSCSEL> to "1" to switch the system clock to the external oscillator.
(8) Confirm CGOSCCR[17]<OSCSEL> become "1". (External oscillator is selected.)
(9) Sets CGOSCCR[16]<XEN2> to "0". (Internal oscillator is disabled.)

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

6.3.6. Prescaler Clock Control

Each peripheral function (TMRB0-7 and SIO0-3) has a prescaler for dividing a clock. As the clock $\varphi T0$ to be input to each prescaler, the "fperiph" clock specified in the CGSYSCR<FPSEL> can be divided according to the setting in the CGSYSCR<PRCK2:0>. After the controller is reset, fperiph is selected as $\varphi T0$.

(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.**

6.4. Modes and Mode Transitions

6.4.1. Mode Transitions

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

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

Fig.6-2 shows a mode transition diagram

For a description of sleep-on-exit, refer to "Cortex-M3 Technical Reference Manual".
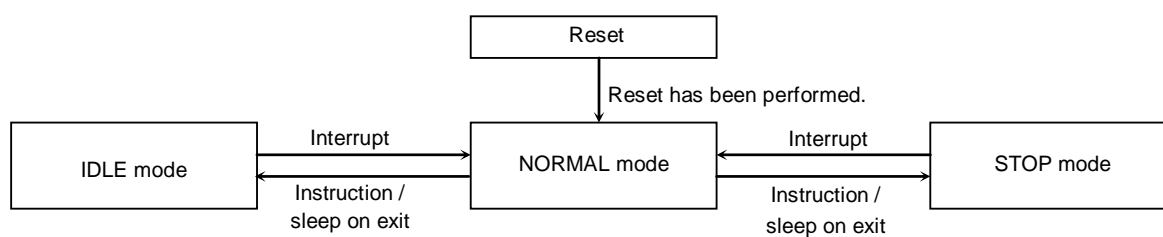
Fig.6-2 Mode Transition Diagram

6.5. Operation Modes

As an operation mode, NORMAL is available. The features of NORMAL mode are described below.

6.5.1. NORMAL Mode

This mode is to operate the CPU core and the peripheral hardware by using the high-speed clock. It is shifted to the NORMAL mode after reset.

6.6. Low Power Consumption Modes

The TMPM372/3/4 has two low power consumption modes: IDLE and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY2:0> 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 chapter 6 for details.

(Note 1)	**Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited as the TMPM372/3/4 does not offer any event for releasing the low power consumption mode.**
(Note 2)	**The TMPM372/3/4 does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the SLEEPDEEP bit of the system control register is prohibited.**

The features of each mode are described as follows.

6.6.1. IDLE Mode

Only the CPU is stopped in this mode.

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

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

- 16-bit timer/event counter (TMRB)
- Serial channel (SIO)
- Watchdog timer (WDT)
- Vector Engine (VE)

6.6.2. STOP Mode

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

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

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>.Table 6-3 shows the pin status in the STOP mode.

(Note1) In STOP modes, the PLL is disabled. When returning from these modes, configure the warm-up time in consideration of the stability time of the PLL. It takes approx. 200μs for the PLL to be stabilized. When the PLL is not used, 60μs or more is needed for warm-up to stabilize the internal circuit.

(Note2) When PB4 is configured as a debug function pin, it prevents the low power consumption mode from being fully effective. Configure PB4 to function as a general-purpose port when the debug function is not used.

Table 6-3 Pin States in STOP Mode

Function		Port	Regsiter setting	I/O	<DRVE>=0	<DRVE>=1
Ports	X2	PM1	CGOSCCR <HOSCON>=1	Output only	"H" level output	
	X1	PM0		Input only	✕	
	TMS TCK TDI $\overline{TRST}$	PB3 PB4 PB6 PB7	PBFR<n>=1 PBIE<n>=1	Input	○	
	TDO	PB5	PBFR<5>=1 PBCR<5>=1	Output	Enabled when data is valid. Disabled when data is invalid.	
	SWCLK	PB4	PBFR<4>=1 PBIE<4>=1	Input	○	
	SWDIO	PB3	PBFR<3>=1 PBCR<3>=1 PBIE<3>=1	Input	○	
				Output	Enabled when data is valid. Disabled when data is invalid.	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	PB0 PB1 PB2 PB5	PBFR<n>=1 PBCR<n>=1	Output	○	
	INT3 INT4 INT5 INT6 INT7 INT8 INTC INTD INTE INTF	PA0 PA2 PE4 PE6 PE7 PA7 PJ6 PJ7 PK0 PK1	PxFR<n>=1 PxIE<n>=1	Input	○	
	Other function pins other than the above or the ports that are used as general purpose ports.		PxIE<n>=1	Input	✕	○
			PxCR<n>=1	Input	✕	○
Except Ports	$\overline{RESET}$,MODE			Input only	○	○

Note 1: TMPM373 doesn't have PA0 and PA7 pins. TMPM374 doesn't have PA0, PA2 and PA7 pins.
Note 2: ○: Input or output enabled   ✕: Input or output disabled.

6.6.3. Low power Consumption Mode Setting

The low power consumption mode is specified by the setting of the standby control register CGSTBYCR<STBY2:0>.

Table 6-4 shows the mode setting in the <STBY2:0>.

Table 6-4 Low power consumption mode setting

Mode	CGSTBYCR<STBY2:0>
STOP	001
IDLE	011

(Note)   Do not set any value other than those shown above in <STBY2:0>.

6.6.4. Operational State in Each Mode

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

For I/O port, "○" and "✕" indicate that input/output is enabled and disabled respectively. For other functions, "○" and "✕" indicate that clock is supplied and is not supplied respectively.

Table 6-5 Operational State in Each Mode

Block	NORMAL	IDLE	STOP
Processor core	○	✕	✕
I/O port	○	○	* (Note 1)
PMD	○	○	✕
ENC	○	○	✕
OFD	○	○	✕
ADC	○	○	✕
VE	○	ON/OFF selectable for each module	✕
SIO	○		✕
TMRB	○		✕
WDT	○		✕
VLTD	○	○	○ (Note 2)
POR	○	○	○ (Note 2)
CG	○	○	✕
PLL	○	○	✕
High-speed oscillator (fc)	○	○	✕

○: Operating, ✕: Stopped

(Note 1)   The state depends on the CGSYSCR<DRVE> bit.

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

6.6.5. Releasing the Low Power Consumption Mode

The low power consumption mode can be released by an interrupt request or reset. The release source that can be used is determined by the low power consumption mode selected. Details are shown in Table 6-6.

Table 6-6 Release Source in Each Mode

Low power consumption mode			IDLE	STOP
Release source	Interrupt	INT3 to INT8, INTC to INTF (Note 1)(Note 2)	○	○
		INTRX0 to 3, INTTX0 to 3 (Note 2)	○	✕
		INTVCNB	○	✕
		INTEMG1	○	✕
		INTOVV1	○	✕
		INTTB00, 01, 10, 11, 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71	○	✕
		INTPMD1	○	✕
		INTENC1	○	✕
		INTCAP00, 01, 10, 11, 20, 21, 30, 31, 40, 41, 50, 60, 61, 70, 71 (Note 2)	○	✕
		INTAD1PDA, INTAD1PDB	○	✕
		INTAD1CPA, INTAD1CPB	○	✕
		INTAD1SFT	○	✕
		INTAD1TMR	○	✕
	NMI (INTWDT)		○	✕
	RESET (RESET pin)		○	○

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

✕:   Unavailable

(Note 1)	**To release the low power consumption mode by using the level mode interrupt, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt handling from starting properly.**
(Note 2)	**The following interrupts in TMPM373 are invalid.**
	INT3, INT8, INTRX2, INTTX2, INTCAP01, INTCAP41
	The following interrupts in TMPM374 are invalid.
	INT3, INT4, INT8, INTRX2, INTTX2, INTCAP01, INTCAP11, INTCAP41
(Note 3)	**For shifting to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified interrupt.**

- Release by interrupt request

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

- Release by Non-Maskable Ininterrupt (NMI)

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

- Release by reset

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

    Note that returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal valid until the oscillator operation becomes stable.

    Refer to "Interrupts" for details.

6.6.6. Warm-up

Mode transition requires the warm-up for stabilization of the internal circuit.

In the mode transition from STOP to NORMAL, the warm-up counter is activated automatically. And then the system clock output is started after the elapse of configured warm-up time. It is necessary to set the oscillator to be used for warm-up in the CGOCCR<WUPSEL> (Note1) and to set the warm-up time in the CGOSCCR<WUODR13:2> before executing the instruction to enter the STOP mode.

(Note1)	**Always set CGOSCCR<WUPSEL1> to "0".**
(Note2)	**In STOP mode, the PLL is disabled. When returning from these modes, configure the warm-up time in consideration of the stability time of the PLL and the internal oscillator. It takes approx. 200µs for the PLL to be stabilized.**

Table 6-7 shows whether the warm-up setting of each mode transition is required or not.

Table 6-7 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL→IDLE	Not required
NORMAL→STOP	Not required
IDLE→NORMAL	Not required
STOP→NORMAL	Auto-warm-up (Note 1)

(Note 1)	**Returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal valid until the oscillator operation becomes stable.**

6.6.6.1. Clock Operations in Mode Transition

The clock operations in mode transition are described in the following section.

6.6.6.1.1. Transition of operation modes: NORMAL→STOP→NORMAL

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

Returning to NORMAL mode by reset does not induce the automatic warm-up. Keep the reset signal asserted until the oscillator operation becomes stable.
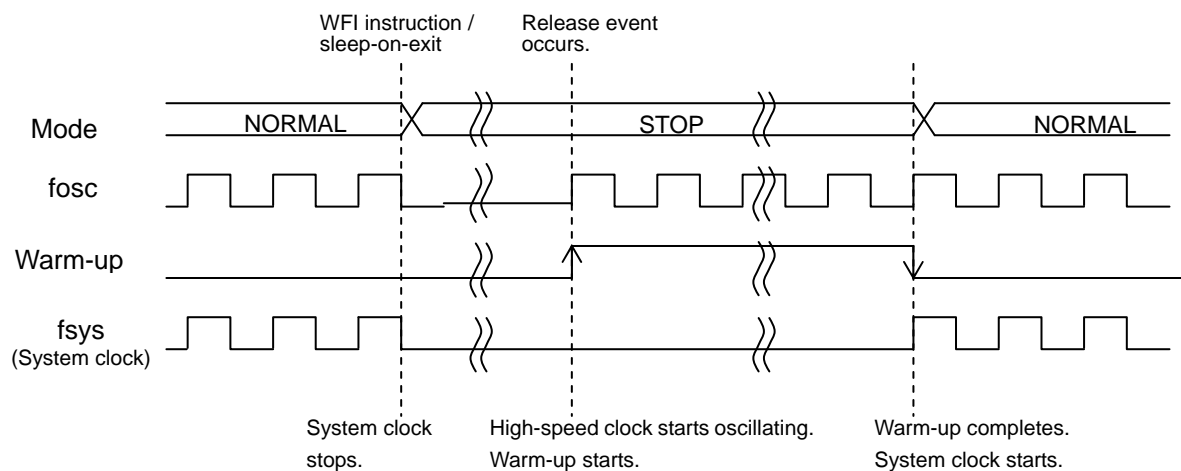
7. Internal High-speed Oscillation Adjustment Function

TMPM372/ 373/ 374 has the internal high-speed oscillation adjustment function.

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

7.1 Structure

The internal oscillation adjustment function uses the pulse width measurement function of 16-bit timer/ event counter (TMRB). Figure 7-1 shows the function configuration.
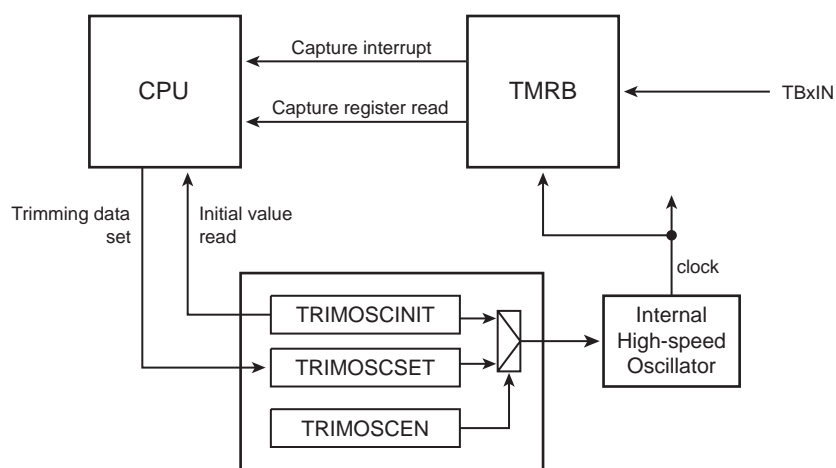
Figure 7-1 Function block diagram

7.2 Registers

7.2.1 Register list

The control registers and its addresses are as follows.

Base Address = 0x4004_0300

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

7.2.2 TRMOSCPRO (Protect register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							

	7	6	5	4	3	2	1	0
bit symbol	PROTECT							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Writing register control 0xC1 : Enable Other than 0xC1 : Desable When "0xC1" is set, TRMOSCEN, TRMOSCINIT and TRMOSCSET are allowed to write.							

7.2.3 TRMOSCEN (Enable register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	TRIMEN
Read/Write	R							R/W
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							Trimming control 0 : Disable 1 : Enable

<TRIMEN> :   When "1" is set, a trimming value of the internal oscillator is switched from a value of TRIMOSCINIT to a value of TRMOSCSET.

7.2.4 TRMOSCINIT (Initial trimming value monitor register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Read as "0".							
	15	14	13	12	11	10	9	8
bit symbol	-	-	TRIMINITC					
Read/Write	R		R					
After reset	0	0	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Function	Read as "0".		Initial coarse trimmng value Enables to monitor initial coarse trimming value.					
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	TRIMINITF			
Read/Write	R				R			
After reset	0	0	0	0	Undefined	Undefined	Undefined	Undefined
Function	Read as "0".				Initial fine trimming value Enables to monitor initial fine trimming value.			

For details about the specific setting and adjustment value of coarse trimming and fine trimming, refer to " 7.3.2 Adjustment range ".

7.2.5   TRMOSCSET (Trimming value setting register)

		31	30	29	28	27	26	25	24
bit symbol		-	-	-	-	-	-	-	-
Read/Write		R							
After reset		0	0	0	0	0	0	0	0
Function		Read as "0".							
		23	22	21	20	19	18	17	16
bit symbol		-	-	-	-	-	-	-	-
Read/Write		R							
After reset		0	0	0	0	0	0	0	0
Function		Read as "0".							
		15	14	13	12	11	10	9	8
bit symbol		-	-	TRIMSETC					
Read/Write		R		R/W					
After reset		0	0	0	0	0	0	0	0
Function		Read as "0".		Coarse trimmng value setting Sets the coarse trimming value.					
		7	6	5	4	3	2	1	0
bit symbol		-	-	-	-	TRIMSETF			
Read/Write		R				R/W			
After reset		0	0	0	0	0	0	0	0
Function		Read as "0".				Fine trimmng value setting Sets the fine trimming value.			

For details about the specific setting and adjustment value of coarse trimming and fine trimming, refer to " 7.3.2 Adjustment range ".

7.3   Operational Description

7.3.1   Outline

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

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

7.3.2   Adjustment range

In the coarse trimming, -57.6% to +55.8% adjustment by 1.8%-step is feasible. In the fine trimming, -2.4% to +2.1% adjustment by 0.3%-step is feasible. Table 7-1 shows a adjustment range.

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

Table 7-1   Adjustment range

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

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

7.3.3   Internal Oscillation Frequency Measurement using TMRB

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

8. 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.

8.1 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.
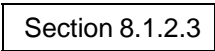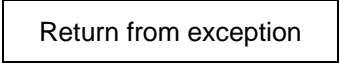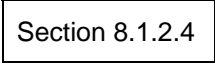
8.1.1 Exception Types
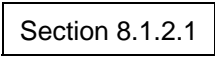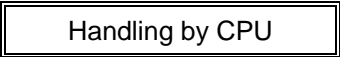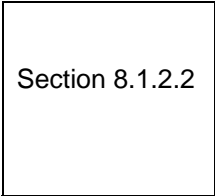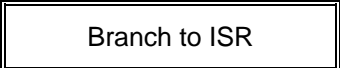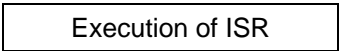
The following types of exceptions exist in the Cortex-M3.

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

8.1.2 **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 8.1.2.1
Handling by CPU	The CPU handles the exception request.	Section 8.1.2.2
Branch to ISR	The CPU branches to the corresponding interrupt service routine (ISR).	
Execution of ISR	Necessary processing is executed。	Section 8.1.2.3
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 8.1.2.4

8.1.2.1 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 "8.5 Interrupts".

(2) Exception detection

If multiple exceptions occur simultaneously, the CPU takes the exception with the highest priority. Table 8-1 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 8-1 Exception Types and Priority

No	Exception type	Priority	Description
1	Reset	-3 (highest)	Reset pin, WDT ,POR, VLTD, OFD or SYSRESETREQ
2	Non-Maskable Interrupt	-2	WDT
3	Hard Fault	-1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4	Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) (Note 1) Instruction fetch from the Execute Never (XN) region
5	Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map
6	Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution
7-10	Reserved		
11	SVCall	Configurable	System service call with SVC instruction
12	Debug Monitor	Configurable	Debug monitor when the core is not halting
13	Reserved		
14	PendSV	Configurable	Pendable system service request
15	SysTick	Configurable	Notification from system timer
16-	External Interrupt	Configurable	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 "8.5.1.5 List of Interrupt Factors".**

(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 of <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 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 8-2 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 8-2 Priority grouping setting

<PRIGROUP[2:0]> setting	<PRI_n[7:0]>		Number of pre-emption priorities	Number of sub priorities
	Pre-emption field	Sub priority field		
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
110	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".**

8.1.2.2 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.

Old SP	<previous>
	xPSR
	PC
	LR
	r12
	r3
	r2
	r1
SP	r0

(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 space. By setting the Vector Table Offset Register, it is possible to 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 are prepared in case 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

8.1.2.3 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 "8.5 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.

8.1.2.4 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 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.

8.2   Reset Exceptions

Reset exceptions are generated from the following six sources.

Use the Reset Flag (RSTFLG) 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".

At least 12 Low level system clock is necessary for reset the device, with the condition that the power

supply is within the operation voltage and high frequency oscilation is stable.

Please refer the "Electrical characteristics" for detail about power supply sequence.

・Reset exception by POR

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

・Reset exception by VLTD

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

・Reset exception by OFD

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

・Reset exception by WDT

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

・Reset exception by SYSRESETREQ

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

Reset Control Register.

8.3   Non-Maskable Interrupts (NMIs)

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

8.4   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 system timer counts based on a clock obtained by dividing the clock selected by CGOSCCR\<OSCSEL\> and \<HOSCON\> into 32.** **The SysTick Calibration Value Register is set to 0x9C4, which provides 10 ms timing when the clock input from X1 is 8 MHz. In case of 10MHz clock input, 10 ms timing is made by setting 0xC35 to SysTick Reload Register.**

8.5   Interrupts

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

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

It sets priority on the 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.

8.5.1   Interrupt factors

8.5.1.1   Interrupt route

Fig8.1 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 (route 1).

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).
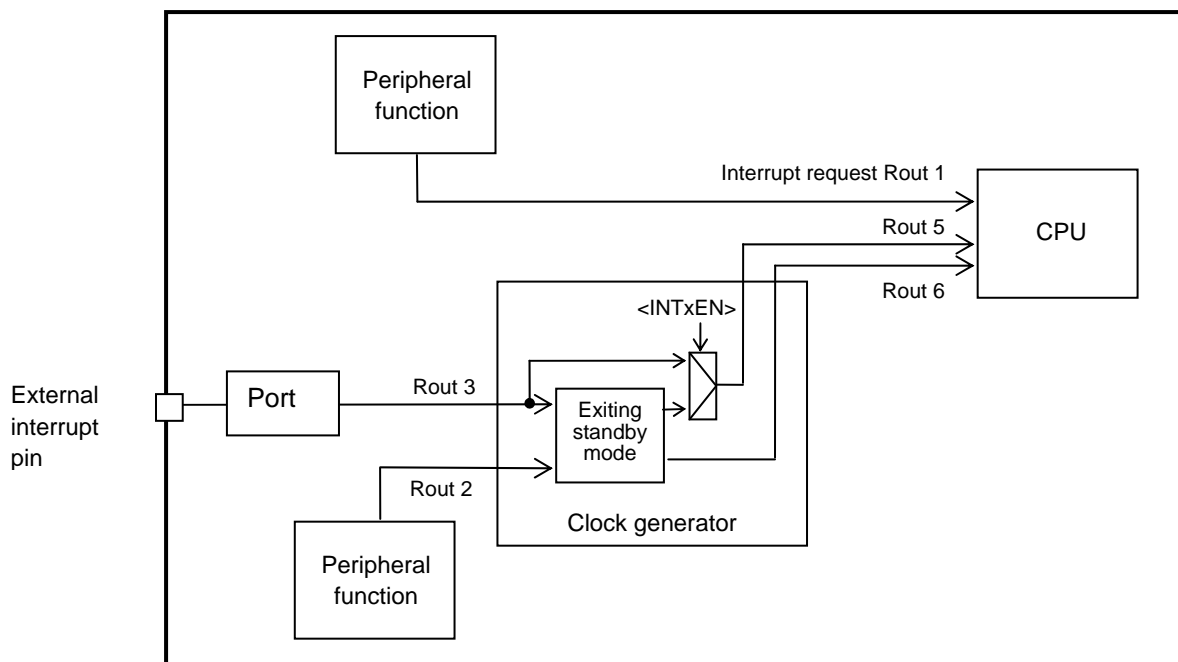
Fig 8.1   Interrupt Route

8.5.1.2 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 chapters 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.

8.5.1.3 Transmission

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

Interrupt requests from interrupt sources that can be used for clearing a standby mode is 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.

8.5.1.4 Precautions when using external interrupt pins

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

If input disabled (PxIE<PnmIE>="0"), inputs from external interrupt pins are "H". Also, if external interrupts are not used as a trigger to release standby (route 6 of "Figure 1-1 Interrupt Route"), input signals from the external interrupt pins are directly sent to the CPU. Since the CPU recognizes "H" 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 "L" and enable it. Then, enable interrupts on the CPU.

8.5.1.5 List of Interrupt Factors

Table 8-3 shows the list of interrupt factors.

Table 8-3 List of Hardware Interrupt Sources (1/2) (○: Available, ✕: non-available)

INT No	Interrupt factors		M372	M373	M374	Active level (Clearing standby)	CG interrupt mode control register
0	Reserved						
1	Reserved						
2	Reserved						
3	INT3	Interrupt Pin (PA0/TB0IN)	○	✕	✕	High/Low Edge/Level Selectable	CGIMCGA
4	INT4	Interrupt Pin (PA2/TB1IN)	○	○	✕		CGIMCGB
5	INT5	Interrupt Pin (PE4/TB2IN)	○	○	○		
6	INTRX0	Serial reception (channel.0)	○	○	○		
7	INTTX0	Serial transmit (channel.0)	○	○	○		
8	INTRX1	Serial reception (channel.1)	○	○	○		
9	INTTX1	Serial transmit (channel.1)	○	○	○		
10	Reserved						
11	INTVCNB	Vector Engine interrupt B	○	○	○		
12	Reserved						
13	INTEMG1	PMD1 EMG interrupt	○	○	○		
14	Reserved						
15	INTOVV1	PMD1 OVV interrupt	○	○	○		
16	Reserved						
17	Reserved						
18	Reserved						
19	INTADBPDB	ADCB conversion triggered by PMD1 is finished	○	○	○		
20	INTTB00	16bit TMRB0 compare match detection 0/ Over flow	○	○	○		
21	INTTB01	16bit TMRB0 compare match detection 1	○	○	○		
22	INTTB10	16bit TMRB1 compare match detection 0/ Over flow	○	○	○		
23	INTTB11	16bit TMRB1 compare match detection 1	○	○	○		
24	INTTB40	16bit TMRB4 compare match detection 0/ Over flow	○	○	○		
25	INTTB41	16bit TMRB4 compare match detection 1	○	○	○		
26	INTTB50	16bit TMRB5 compare match detection 0/ Over flow	○	○	○		
27	INTTB51	16bit TMRB5 compare match detection 1	○	○	○		
28	Reserved						
29	INTPMD1	PMD1 PWM interrupt	○	○	○		
30	INTCAP00	16bit TMRB0 input capture 0	○	○	○		
31	INTCAP01	16bit TMRB0 input capture 1	○	✕	✕		
32	INTCAP10	16bit TMRB1 input capture 0	○	○	○		
33	INTCAP11	16bit TMRB1 input capture 1	○	○	✕		
34	INTCAP40	16bit TMRB4 input capture 0	○	○	○		
35	INTCAP41	16bit TMRB4 input capture 1	○	✕	✕		
36	INTCAP50	16bit TMRB5 input capture 0	○	○	○		
37	Reserved						
38	INT6	Interrupt Pin (PE6/TB3IN)	○	○	○	High/Low Edge/Level Selectable	CGIMCGB
39	INT7	Interrupt Pin (PE7/TB3OUT)	○	○	○		

Table 8-4 List of Hardware Interrupt Sources (2/2) (○: Available, ✕: non-available)

No.		Interrupt factors	M372	M373	M374	Active level (Clearing standby)	CG interrupt mode control register
40	INTRX2	Serial reception (channel.2)	○	✕	✕		
41	INTTX2	Serial transmit (channel.2)	○	✕	✕		
42	Reserved						
43	INTADBCPA	ADCB conversion monitoring function interrupt A	○	○	○		
44	Reserved						
45	INTADBCPB	ADCB conversion monitoring function interrupt B	○	○	○		
46	INTTB20	16bit TMRB2 compare match detection 0/ Over flow	○	○	○		
47	INTTB21	16bit TMRB2 compare match detection 1	○	○	○		
48	INTTB30	16bit TMRB3 compare match detection 0/ Over flow	○	○	○		
49	INTTB31	16bit TMRB3 compare match detection 1	○	○	○		
50	INTCAP20	16bit TMRB2 input capture 0	○	○	○		
51	INTCAP21	16bit TMRB2 input capture 1	○	○	○		
52	INTCAP30	16bit TMRB3 input capture 0	○	○	○		
53	INTCAP31	16bit TMRB3 input capture 1	○	○	○		
54	Reserved						
55	INTADBSFT	ADCB conversion started by software is finished	○	○	○		
56	Reserved						
57	INTADBTMR	ADCB conversion triggered by timer is finished	○	○	○		
58	INT8	Interrupt Pin (PA7/TB4IN)	○	✕	✕	High/Low Edge/Level Selectable	CGIMCGC
59	Reserved						
60	Reserved						
61	Reserved						
62	Reserved						
63	INTENC1	Encoder input1 interrupt	○	○	○		
64	INTRX3	Serial reception (channel.3)	○	○	○		
65	INTTX3	Serial transmit (channel.3)	○	○	○		
66	INTTB60	16bit TMRB6 compare match detection 0 / Over flow	○	○	○		
67	INTTB61	16bit TMRB6 compare match detection 1	○	○	○		
68	INTTB70	16bit TMRB7 compare match detection 0 / Over flow	○	○	○		
69	INTTB71	16bit TMRB7 compare match detection 1	○	○	○		
70	INTCAP60	16bit TMRB6 input capture 0	○	○	○		
71	INTCAP61	16bit TMRB6 input capture 1	○	○	○		
72	INTCAP70	16bit TMRB7 input capture 0	○	○	○		
73	INTCAP71	16bit TMRB7 input capture 1	○	○	○		
74	INTC	Interrupt Pin (PJ6/AINB9)	○	○	○	High/Low Edge/Level Selectable	CGIMCGD
75	INTD	Interrupt Pin (PJ7/AINB10)	○	○	○		
76	INTE	Interrupt Pin (PK0/AINB11)	○	○	○		
77	INTF	Interrupt Pin (PK1/AINB12)	○	○	○		

8.5.1.6 Active Level

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

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

If an interrupt source is used for clearing a standby mode, setting the relevant clock generator register is also required. Enable the CGIMCGx<INTxEN> bit and specify the active level in the CGIMCGx<EMCGx> bits. You must set the active level for interrupt requests from each peripheral function as shown in Table 8-3.

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

8.5.2   Interrupt handling

8.5.2.1   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 NVIC registers for detecting interrupts.
	Set the clock generator as well if the interrupt source is used to clear a standby mode.
	○ Common setting
	  NVIC registers
	○ Setting to clear standby mode
	  Clock generator
		8.5.2.2 Preparation

Settings for generating interrupt request signal	Execute an appropriate setting to send generate the interrupt signal depending on the interrupt type.
	○Setting for interrupt from the external pin
	  Port
	○Setting for interrupt from peripheral function
	  Peripheral function (See the chapters of each peripheral function for details.)	

Interrupt generation — An interrupt request is generated.

Not clearing standby mode / Clearing standby mode

| CG detects interrupt (clearing standby mode) | Interrupt lines used for clearing a standby mode are connected to the CPU via the clock generator. | 8.5.2.3 Detection by Clock Generator |

Processing	Details	See

	The CPU detects the interrupt.	8.5.2.4
CPU detects interrupt	If multiple interrupt requests occur simultaneously,, the interrupt request with the highest priority is selected according to the priority order.	Detection by CPU

CPU handles interrupt	The CPU handles the interrupt. The CPU pushes resister contents to the stack before entering the ISR	8.5.2.5 CPU processing

ISR Execution	Program for the ISR. Clear the interrupt factor if needed.	8.5.2.6 Interrupt service routine

Returning to preceding program	Configure to return to the preceding program of the ISR.	

8.5.2.2 Preparation

When preparing for an interrupt, it is needed 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)

(Note1)	PRIMASK register cannot be modified by the user access level.
(Note2)	If a fault causes when "1" is set to the PRIMASK register, it is treated as a hard fault.

（2）CPU interrupt registers setting

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

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 PnFRx[m] allows the pin to be used as the function pin. Setting PnIE[m] allows the pin to be used as the input port.

● Port register		
PnFRx<PnmFRx>	←	"1"
PnIE<PnmIE>	←	"1"

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

（4）Preconfiguration 2 (interrupt from peripheral function)

The setting varies depending on the peripheral function to be used. See chapters 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 registers 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 "8.6.3.5 CGICRCG (CG Interrupt 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 "H" pulse or "H" level signal must be input so that the CPU can detect it as an interrupt request.. Also, be aware of the description of "8.5.1.4 Precautions when using external interrupt pins".

●Clock generator register		
CGIMCGn<EMCGm>	←	Active level
CGICRCG<ICRCG4-0>	←	Value corresponding to the interrupt to be used
CGIMCGn<INTmEN>	←	"1" (interrupt enabled)

(Note)	n: register number
	m: number assigned to each interrupt factor

（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"

(Note1)	m: corresponding bit
(Note2)	PRIMASK register cannot be modified by the user access level.

8.5.2.3 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.

8.5.2.4 Detection by CPU

The CPU detects an interrupt factor with the highest priority.

8.5.2.5 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.

8.5.2.6 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 factor 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 with the 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 interrupt factor

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.

8.6 Exception/Interrupt-related registers

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

8.6.1 Register List

●NVIC Resisters	
SysTick Control and Status Resister	0xE000_E010
SysTick Reload Value Resister	0xE000_E014
SysTick Current Value Resister	0xE000_E018
SysTick Calibration Value Register	0xE000_E01C
Interrupt Set-Enable Register 1	0xE000_E100
Interrupt Set-Enable Register 2	0xE000_E104
Interrupt Set-Enable Register 3	0xE000_E108
Interrupt Clear-Enable Register 1	0xE000_E180
Interrupt Clear-Enable Register 2	0xE000_E184
Interrupt Clear-Enable Register 3	0xE000_E188
Interrupt Set-Pending Register 1	0xE000_E200
Interrupt Set-Pending Register 2	0xE000_E204
Interrupt Set-Pending Register 3	0xE000_E208
Interrupt Clear-Pending Register 1	0xE000_E280
Interrupt Clear-Pending Register 2	0xE000_E284
Interrupt Clear-Pending Register 3	0xE000_E288
Interrupt Priority Register	0xE000_E400-0xE000_E450
Vector Table Offset Register	0xE000_ED08
Application Interrupt and Reset Control Register	0xE000_ED0C
System Handler Priority Register	0xE000_ED18,0xE000_ED1C,0xE000_ED20
System Handler Control and State Register	0xE000_ED24

●Clock generator registers		
CGICRCG	CG Interrupt Request Clear Register	0x4004_0214
CGNMIFLG	NMI Flag Register	0x4004_0218
CGRSTFLG	Reset Flag Register	0x4004_021C
CGIMCGA	CG Interrupt Mode Control Register A	0x4004_0220
CGIMCGB	CG Interrupt Mode Control Register B	0x4004_0224
CGIMCGC	CG Interrupt Mode Control Register C	0x4004_0228
CGIMCGD	CG Interrupt Mode Control Register D	0x4004_022C
Reserved	-	0x4004_0230
Reserved	-	0x4004_0234
Reserved	-	0x4004_0238
Reserved	-	0x4004_023C

(Note)   Access to the "Reserved" areas is prohibited.

8.6.2   NVIC Registers

8.6.2.1 SysTick Control and Status Register

	7	6	5	4	3	2	1	0
bit Symbol						CLK SOURCE	TICKINT	ENABLE
Read/Write	R					R/W	R/W	R/W
After reset	0					0	0	0
Function	"0" is read.					0: External reference clock 1: Core clock	0: Do not pend SysTick 1: Pend SysTick	0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								COUNT FLAG
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							0: Timer not counted to 0 1: Timer counted to 0
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit0>   <ENABLE>   1 = The counter loads with the Reload value and then begins counting down.
                      0 = The timer is disabled.

<bit1>   <TICKINT>   1 = SysTick exceptions are pended.
                       0 = SysTick exceptions are not pended.

<bit2>   <CLKSOURCE>   0 = External reference clock
                        1 = Core clock

<bit16>   <COUNTFLAG>   1 =Indicates that the timer counted to 0 since last time this was read.
                        Clears on read of any part of the SysTick Control and Status Register.

8.6.2.2 SysTick Reload Value Register

	7	6	5	4	3	2	1	0
bit Symbol	RELOAD							
Read/Write	R/W							
After reset	Undefined							
Function	Reload value							

	15	14	13	12	11	10	9	8
bit Symbol	RELOAD							
Read/Write	R/W							
After reset	Undefined							
Function	Reload value							

	23	22	21	20	19	18	17	16
bit Symbol	RELOAD							
Read/Write	R/W							
After reset	Undefined							
Function	Reload value							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit23:0> <RELOAD> Set the value to load into the SysTick Current Value Register when the timer reaches "0".

(Note) In this product, the system timer counts based on a clock obtained by dividing the clock selected by CGOSCCR<OSCSEL> and <HOSCON> into 32.

8.6.2.3 SysTick Current Value Register

	7	6	5	4	3	2	1	0
bit Symbol	CURRENT							
Read/Write	R/W							
After reset	Undefined							
Function	[Read] Current SysTick timer value [Write] Clear							
	15	14	13	12	11	10	9	8
bit Symbol	CURRENT							
Read/Write	R/W							
After reset	Undefined							
Function	[Read] Current SysTick timer value [Write] Clear							
	23	22	21	20	19	18	17	16
bit Symbol	CURRENT							
Read/Write	R/W							
After reset	Undefined							
Function	[Read] Current SysTick timer value [Write] Clear							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit23:0>   <CURRENT>   [Read] Current SysTick timer value.

[Write] 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.

8.6.2.4 SysTick Calibration Value Register

	7	6	5	4	3	2	1	0
bit Symbol	TENMS							
Read/Write	R							
After reset	1	1	0	0	0	1	0	0
Function	Calibration value (Note)							

	15	14	13	12	11	10	9	8
bit Symbol	TENMS							
Read/Write	R							
After reset	0	0	0	0	1	0	0	1
Function	Calibration value (Note)							

	23	22	21	20	19	18	17	16
bit Symbol	TENMS							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Calibration value (Note)							

	31	30	29	28	27	26	25	24
bit Symbol	NOREF	SKEW						
Read/Write	R	R	R					
After reset	0	0	0					
Function	0: Reference clock provided 1: No reference clock	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.	"0" is read.					

<bit23:0>   <TENMS>   Reload value to use for 10 ms timing (0x9C4). (Note)

<bit30>   <SKEW>   1 = The calibration value is not exactly 10 ms.

<bit31>   <NOREF>   1 = The reference clock is not provided.

> **(Note)**   In this product, the system timer counts based on a clock obtained by dividing the clock selected by CGOSCCR<OSCSEL> and <HOSCON> into 32.
> The SysTick Calibration Value Register is set to a value that provides 10 ms timing when the cock input from X1 is 8 MHz.   In case of 10MHz clock input, 10 ms timing is made by setting 0xC35 to SysTick Reload Register.

8.6.2.5 Interrupt Set-Enable Register 1

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 7 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 6 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 5 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 4 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 3 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".		

	15	14	13	12	11	10	9	8
bit Symbol	SETENA		SETENA		SETENA		SETENA	
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 15 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 13 [Write] 1: Enable [Read] 0 Disabled 1: Enabled	Write "0".	Interrupt number 11 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 9 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 8 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 23 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 22 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 21 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 20 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 19 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".		

	31	30	29	28	27	26	25	24
bit Symbol	SETENA				SETENA			
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 31 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 30 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 29 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 27 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 26 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 25 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 24 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

<bit31:0> <SETENA>   Use these bits to enable interrupts or determine which interrupts are currently enabled.
Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.
Reading a bit in this register returns the current state of the corresponding interrupt as shown below.
0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 8.5.1.5 List of Interrupt Factors.

8.6.2.6 Interrupt Set-Enable Register 2

	7	6	5	4	3	2	1	0
bit Symbol	SETENA			SETENA				
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 39 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 38 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 36 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 35 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 34 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 33 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 32 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol	SETENA				SETENA		SETENA	
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 47 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 46 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 45 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 43 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 41 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 40 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	SETENA		SETENA					
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 55 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 53 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 52 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 51 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 50 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 49 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 48 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	SETENA					SETENA		
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 63 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".				Interrupt number 58 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 57 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Write "0".

<bit31:0>   <SETENA>   Use these bits to enable interrupts or determine which interrupts are currently enabled.
Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.
Reading a bit in this register returns the current state of the corresponding interrupt as shown below.
0 = Disabled
1 = Enabled

(Note)   For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.7 Interrupt Set-Enable Register 3

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 71 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 70 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 69 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 68 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 67 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 66 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 65 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 64 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	15	14	13	12	11	10	9	8
bit Symbol			SETENA					
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.		Interrupt number 77 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 76 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 75 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 74 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 73 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 72 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

<bit13:0> <SETENA> Use these bits to enable interrupts or determine which interrupts are currently enabled.

Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.8 Interrupt Clear-Enable Register 1

	7	6	5	4	3	2	1	0
bit Symbol			CLRENA					
Read/Write				R/W				
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 7 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 6 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 5 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 4 [Write] 1: Disable [read] 0: Disabled 1: Enabled	Interrupt number 3 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".		

	15	14	13	12	11	10	9	8
bit Symbol	CLRENA		CLRENA		CLRENA		CLRENA	
Read/Write				R/W				
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 15 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 13 [Write] 1: Disable [Read] 0: Disabled 1 Enabled	Write "0".	Interrupt number 11 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 9 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 8 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol			CLRENA					
Read/Write				R/W				
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 23 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 22 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 21 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 20 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 19 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".		

	31	30	29	28	27	26	25	24
bit Symbol		CLRENA				CLRENA		
Read/Write				R/W				
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 31 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 30 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 29 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 27 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 26 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 25 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 24 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

<bit31:0> <CLRENA> Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note) **For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.**

8.6.2.9 Interrupt Clear-Enable Register 2

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA			CLRENA				
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 39 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 38 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 36 [Write] 1: Disable [read] 0: Disabled 1: Enabled	Interrupt number 35 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 34 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 33 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 32 [Write] 1: Disable [Read] 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol	CLRENA				CLRENA		CLRENA	
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 47 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 46 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 45 [Write] 1: Disable [Read] 0: Disabled 1 Enabled	Write "0".	Interrupt number 43 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 41 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 40 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol	CLRENA		CLRENA					
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 55 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".	Interrupt number 53 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 52 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 51 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 50 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 49 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 48 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	31	30	29	28	27	26	25	24
bit Symbol	CLRENA					CLRENA		
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 63 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".				Interrupt number 58 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 57 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Write "0".

<bit31:0> <CLRENA> Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note)   For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.10 Interrupt Clear-Enable Register 3

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 71 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 70 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 69 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 68 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 67 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 66 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 65 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 64 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	15	14	13	12	11	10	9	8
bit Symbol			CLRENA					
Read/Write			R/W					
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.		Interrupt number 77 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number76 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 75 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 74 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 73 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 72 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit13:0>   <CLRENA>   Use these bits to disable or determine which interrupts are currently disabled.

Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect.

Reading a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.11 Interrupt Set-Pending Register 1

	7	6	5	4	3	2	1	0
bit Symbol			SETPEND					
Read/Write				R/W				
After reset			Undefined			0	0	0
Function	Interrupt number 7 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 6 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 5 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 4 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 3 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".		

	15	14	13	12	11	10	9	8
bit Symbol	SETPEND		SETPEND		SETPEND		SETPEND	
Read/Write				R/W				
After reset	Undefined	0	Undefined	0	Undefined	0	Undefined	
Function	Interrupt number 15 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 13 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 11 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 9 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 8 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol			SETPEND					
Read/Write				R/W				
After reset			Undefined			0	0	0
Function	Interrupt number 23 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 22 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 21 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 20 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 19 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".		

	31	30	29	28	27	26	25	24
bit Symbol		SETPEND				SETPEND		
Read/Write				R/W				
After reset		Undefined		0		Undefined		
Function	Interrupt number 31 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 30 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 29 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 27 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 26 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 25 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 24 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit31:0> <SETPEND> Use these bits to force interrupts into the pending state or determine 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 a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note) For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.12 Interrupt Set-Pending Register 2

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND			SETPEND				
Read/Write	R/W							
After reset	Undefined		0	Undefined				
Function	Interrupt number 39 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 36 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	SETPEND				SETPEND		SETPEND	
Read/Write	R/W							
After reset	Undefined			0	Undefined	0	Undefined	
Function	Interrupt number 47 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 46 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 45 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 43 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 41 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number40 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	SETPEND		SETPEND					
Read/Write	R/W							
After reset	Undefined	0	Undefined					
Function	Interrupt number 55 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 53 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 52 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 51 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 50 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 49 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 48 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	SETPEND					SETPEND		
Read/Write	R/W							
After reset	Undefined	0	0	0	0	Undefined		0
Function	Interrupt number 63 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".				Interrupt number 58 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 57 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Write "0".

<bit31:0>   <SETPEND>   Use these bits to force interrupts into the pending state or determine 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 a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note)   **For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.**

8.6.2.13 Interrupt Set-Pending Register 3

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 71 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 70 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 69 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 68 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 67 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 66 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 65 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 64 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol			SETPEND					
Read/Write	R		R/W					
After reset	0		Undefined					
Function	"0" is read.		Interrupt number 77 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 76 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 75 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 74 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 73 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 72 [Write] 1: Pend [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit13:0>   <SETPEND>   Use these bits to force interrupts into the pending state or determine 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 a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

Each bit in this register can be cleared by writing "1" to the corresponding bit in the Interrupt Clear-Pending Register.

(Note)   **For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.**

8.6.2.14 Interrupt Clear-Pending Register 1

	7	6	5	4	3	2	1	0
bit Symbol			CLRPEND					
Read/Write				R/W				
After reset			Undefined			0	0	0
Function	Interrupt number 7 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 6 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 5 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 4 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 3 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".		

	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND		CLRPEND		CLRPEND		CLRPEND	
Read/Write				R/W				
After reset	Undefined	0	Undefined	0	Undefined	0	Undefined	
Function	Interrupt number 15 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 13 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 11 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 9 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 8 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol			CLRPEND					
Read/Write				R/W				
After reset			Undefined			0	0	0
Function	Interrupt number 23 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 22 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 21 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 20 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 19 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".		

	31	30	29	28	27	26	25	24
bit Symbol		CLRPEND				CLRPEND		
Read/Write				R/W				
After reset		Undefined		0			Undefined	
Function	Interrupt number 31 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 30 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 29 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 27 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 26 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 25 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 24 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

<bit31:0> <CLRPEND> Use these bits to clear pending interrupts or determine 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 a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note) For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.15 Interrupt Clear-Pending Register 2

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND			CLRPEND				
Read/Write	R/W							
After reset	Undefined		0	Undefined				
Function	Interrupt number39 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 36 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND				CLRPEND		CLRPEND	
Read/Write	R/W							
After reset	Undefined			0	Undefined	0	Undefined	
Function	Interrupt number 47 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 46 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 45 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 43 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 41 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 40 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol	CLRPEND		CLRPEND					
Read/Write	R/W							
After reset	Undefined	0	Undefined					
Function	Interrupt number 55 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".	Interrupt number 53 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 52 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 51 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 50 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 49 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 48 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	31	30	29	28	27	26	25	24
bit Symbol	CLRPEND					CLRPEND		
Read/Write	R/W							
After reset	Undefined	0	0	0	0	Undefined		0
Function	Interrupt number 63 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".				Interrupt number58 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 57 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Write "0".

<bit31:0> <CLRPEND> Use these bits to clear pending interrupts or determine 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 a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note) For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.16 Interrupt Clear-Pending Register 3

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 39 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 37 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 36 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	15	14	13	12	11	10	9	8
bit Symbol			CLRPEND					
Read/Write	R		R/W					
After reset	0		Undefined					
Function	"0" is read.		Interrupt number 45 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 44 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 43 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 42 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 41 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 40 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit13:0> <CLRPEND> Use these bits to clear pending interrupts or determine 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 a bit in this register returns the current state of the corresponding interrupt as shown below.

0 = Not pending
1 = Pending

(Note) For descriptions of interrupts and interrupt numbers, see 8.5.1.5 List of Interrupt Factors.

8.6.2.17 Interrupt Priority Registers

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	-	-	-
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	-	PRI_9	PRI_8
0xE000_E40C	PRI_15	-	PRI_13	-
0xE000_E410	PRI_19	-	-	-
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	-
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000_E424	PRI_39	PRI_38	-	PRI_36
0xE000_E428	PRI_43	-	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	-
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E434	PRI_55	-	PRI_53	PRI_52
0xE000_E438	-	PRI_58	PRI_57	-
0xE000_E43C	PRI_63	-	-	-
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_77	PRI_76

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 4 to 7. 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.

	7	6	5	4	3	2	1	0
bit Symbol	PRI_4							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of interrupt number 4			"0" is read.				

	15	14	13	12	11	10	9	8
bit Symbol	PRI_5							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of interrupt number 5			"0" is read.				

	23	22	21	20	19	18	17	16
bit Symbol	PRI_6							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of interrupt number 6			"0" is read.				

	31	30	29	28	27	26	25	24
bit Symbol	PRI_7							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of interrupt number 7			"0" is read.				

<bit7:5>   <PRI_0>   Priority of interrupt number 4

<bit15:13>   <PRI_1>   Priority of interrupt number 5

<bit23:21>   <PRI_2>   Priority of interrupt number 6

<bit31:29>   <PRI_3>   Priority of interrupt number 7

8.6.2.18   Vector Table Offset Register

	7	6	5	4	3	2	1	0
bit Symbol	TBLOFF							
Read/Write	R/W	R						
After reset	0	0						
Function	Offset value	"0" is read.						
	15	14	13	12	11	10	9	8
bit Symbol	TBLOFF							
Read/Write	R/W							
After reset	0							
Function	Offset value							
	23	22	21	20	19	18	17	16
bit Symbol	TBLOFF							
Read/Write	R/W							
After reset	0							
Function	Offset value							
	31	30	29	28	27	26	25	24
bit Symbol			TBLBASE	TBLOFF				
Read/Write	R		R/W	R/W				
After reset	0		0	0				
Function	"0" is read.		Table base	Offset value				

<bit28:7>   <TBLOFF>   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, it is necessary to adjust the alignment by rounding up to the next power of two.

<bit29>   <TBLBASE>   The vector table is in:
0 = Code space
1 = SRAM space

8.6.2.19 Application Interrupt and Reset Control Register

	7	6	5	4	3	2	1	0
bit Symbol						SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
Read/Write			R			R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function			"0" is read.			System Reset Request.	Clear active vector bit	System Reset bit
	15	14	13	12	11	10	9	8
bit Symbol	ENDIAN ESS						PRIGROUP	
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Endianne ss bit		"0" is read.			Interrupt priority grouping		
	23	22	21	20	19	18	17	16
bit Symbol	VECTKEY/VECTKEYSTAT							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05							
	31	30	29	28	27	26	25	24
bit Symbol	VECTKEY/VECTKEYSTAT							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05							

<bit0>　　<VECTRESET>　　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.

<bit1>　　<VECTCLRACTIVE>　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.

<bit2>　　<SYSRESETREQ>　System Reset Request.
　　　　　　　　　　　　　　　1=CPU outputs a SYSRESETREQ signal. (note2)

<bit10:8>　<PRIGROUP>　　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.

<bit15>　　<ENDIANESS>　　Endianness bit:(Note1)

		1: big endian 0: little endian
<bit31:16>	VECTKEY/ VECTSTAT	Register key [Write] Writing to this register requires 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05.

> **(Note1)** Little-endian is the default memory format for this product.
> **(Note2)** When SYSRESETREQ is output, warm reset is performed on this product. <SYSRESETREQ> is cleared by warm reset.

8.6.2.20 System Handler Priority Registers

System Handler Priority Registers have eight bits per each exception.

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. The System Handler Priority Registers for all other exceptions have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.

	7	6	5	4	3	2	1	0
bit Symbol	PRI_4							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Memory Management			"0" is read.				

	15	14	13	12	11	10	9	8
bit Symbol	PRI_5							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Bus Fault			"0" is read.				

	23	22	21	20	19	18	17	16
bit Symbol	PRI_6							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Usage Fault			"0" is read.				

	31	30	29	28	27	26	25	24
bit Symbol	PRI_7							
Read/Write	R/W			R				
After reset	0			0				
Function	Reserved			"0" is read.				

8.6.2.21   System Handler Control and State Register

	7	6	5	4	3	2	1	0
bit Symbol	SVCALL ACT				USGFAULT ACT		BUSFAULT ACT	MEMFAULT ACT
Read/Write	R/W	R			R/W	R	R/W	R/W
After reset	0	0			0	0	0	0
Function	SVCall 0: Inactive 1: Active	"0" is read.			Usage fault 0: Inactive 1: Active	"0" is read.	Bus fault 0: Inactive 1: Active	Memory Management 0: Inactive 1: Active
	15	14	13	12	11	10	9	8
bit Symbol	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICK ACT	PENDSV ACT		MONITOR ACT
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	SVCall 0: Not pended 1: Pended	Bus Fault 0: Not pended 1: Pended	Memory Management 0: Not pended 1: Pended	Usage Fault 0: Not pended 1: Pended	SysTick 0: Inactive 1: Active	PendSV 0: Inactive 1: Active	"0" is read.	Debug Monitor 0: Inactive 1: Active
	23	22	21	20	19	18	17	16
bit Symbol						USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
Read/Write	R					R/W	R/W	R/W
After reset	0					0	0	0
Function	"0" is read.					Usage Fault 0: Disable 1: Enable	Bus Fault 0: Disable 1: Enable	Memory Management 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

<bit0>	<MEMFAULTACT>	Reads as "1" if Memory Management is active.
<bit1>	<BUSFAULTACT>	Reads as "1" if Bus Fault is active.
<bit3>	<USGFALTACT>	Reads as "1" if Usage Fault is active.
<bit7>	<SVCALLACT>	Reads as "1" if SVCall is active.
<bit8>	<MONITORACT>	Reads as "1" if Debug Monitor is active.
<bit10>	<PENDSVACT>	Reads as "1" if PendSV is active.
<bit11>	<SYSTICKACT>	Reads as "1" if SysTick is active.
<bit12>	<USGFAULTPENDED>	Reads as "1" if Usage Fault is pended.
<bit13>	<MEMFAULTPENDED>	Reads as "1" if Memory Management is pended.
<bit14>	<BUSFAULTPENDED>	Reads as "1" if Bus Fault is pended.
<bit15>	<SVCALLPENDED>	Reads as "1" if SVCall is pended.

<bit16>   <MEMFAULTENA>   Set to "0" to disable or "1" to enable Memory Management.

<bit17>   <BUSFAULTENA>   Set to "0" to disable or "1" to enable Bus Fault.

<bit18>   <USGFAULTENA>   Set to "0" to disable or "1" to enable Usage Fault.

(Note)   **Extreme caution is needed to clear or set the active bits, because clearing and setting these bits does not repair stack contents.**

8.6.3 Clock Generator Registers

8.6.3.1 CG Interrupt Mode Control Register A

This resister set the clearing standby request active level of external interrupt INT3.

CGIMCGA		7	6	5	4	3	2	1	0
	bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Write as "010".			"0" is read		"0" is read	Write as "0".

		15	14	13	12	11	10	9	8
	bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Write as "010".			"0" is read		"0" is read	Write as "0".

		23	22	21	20	19	18	17	16
	bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Write as "010".			"0" is read		"0" is read	Write as "0".

		31	30	29	28	27	26	25	24
	bit Symbol	-	EMCG32	EMCG31	EMCG30	EMST31	EMST30	INT3F	INT3EN
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INT3 standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT3 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INT3 clear input 0: Disable 1: Enable

(Note1) EMSTxx is effective only when EMCGxx is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring EMSTxx. If interrupts are cleared with the CGICRCG register, EMSTxx is also cleared.

(Note2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

(Note3) Do not access to CGIMCGA register in case of TMPM373/374.

8.6.3.2   CG Interrupt Mode Control Register B

This resister set the clearing standby request active level of external interrupt INT4 to INT7.

CGIMCGB		7	6	5	4	3	2	1	0
	bit Symbol	-	EMCG42	EMCG41	EMCG40	EMST41	EMST40	INT4F	INT4EN
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INT4 standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT4 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INT4 clear input 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
	bit Symbol	-	EMCG52	EMCG51	EMCG50	EMST51	EMST50	INT5F	INT5EN
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INT5 standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT5 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INT5 clear input 0: Disable 1: Enable
		23	22	21	20	19	18	17	16
	bit Symbol	-	EMCG62	EMCG61	EMCG60	EMST61	EMST60	INT6F	INT6EN
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INT6 standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT6 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INT6 clear input 0: Disable 1: Enable
		31	30	29	28	27	26	25	24
	bit Symbol	-	EMCG72	EMCG71	EMCG70	EMST71	EMST70	INT7F	INT7EN
	Read/Write	R	R/W			R		R	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INT7 standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT7 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INT7 clear input 0: Disable 1: Enable

(Note1)	EMSTxx is effective only when EMCGxx is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring EMSTxx. If interrupts are cleared with the CGICRCG register, EMSTxx is also cleared.
(Note2)	Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.
(Note3)	Always write "0" to bit0 in case of TMPM374.

8.6.3.3   CG Interrupt Mode Control Register C

This resister set the clearing standby request active level of external interrupt INT8.

CGIMCGC

	7	6	5	4	3	2	1	0
bit Symbol	-	EMCG82	EMCG81	EMCG80	EMST81	EMST80	INT8F	INT8EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read	Active state setting of INT8 standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT8 standby clear request 00: ─ 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INT8 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read	Write as "010".			"0" is read		"0" is read	Write as "0".

	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read	Write as "010".			"0" is read		"0" is read	Write as "0".

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read	Write as "010".			"0" is read		"0" is read	Write as "0".

(Note1) EMSTxx is effective only when EMCGxx is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring EMSTxx. If interrupts are cleared with the CGICRCG register, EMSTxx is also cleared.

(Note2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

(Note3) Do not access to CGIMCGC register in case of TMPM373/374.

8.6.3.4   CG Interrupt Mode Control Register D

This resister set the clearing standby request active level of external interrupt INTC to INTF.

CGIMCGD		7	6	5	4	3	2	1	0
	bit Symbol		EMCGC2	EMCGC1	EMCGC0	EMSTC1	EMSTC0	INTCF	INTCEN
	Read/Write	R	R/W			R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INTC standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTC standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INTC clear input 0: Disable 1: Enable
		15	14	13	12	11	10	9	8
	bit Symbol		EMCGD2	EMCGD1	EMCGD0	EMSTD1	EMSTD0	INTDF	INTDEN
	Read/Write	R	R/W			R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INTD standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTD standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INTD clear input 0: Disable 1: Enable
		23	22	21	20	19	18	17	16
	bit Symbol		EMCGE2	EMCGE1	EMCGE0	EMSTE1	EMSTE0	INTEF	INTEEN
	Read/Write	R	R/W			R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INTE standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTE standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INTE clear input 0: Disable 1: Enable
		31	30	29	28	27	26	25	24
	bit Symbol		EMCGF2	EMCGF1	EMCGF0	EMSTF1	EMSTF0	INTFF	INTFEN
	Read/Write	R	R/W			R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function	"0" is read	Active state setting of INTF standby clear request. (101 to 111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTF standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		Interrupt happened 0:No 1:happened	INTF clear input 0: Disable 1: Enable

(Note1)	EMSTxx is effective only when EMCGxx is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring EMSTxx. If interrupts are cleared with the CGICRCG register, EMSTxx is also cleared.
(Note2)	Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

Be sure to set active state of the Standby clear request, in case the interrupt is enabled for clearing the Standby modes.

(Note 1) When using interrupts, be sure to follow the sequence of actions shown below:

      1) If the external intrrupt request pin shared with other general ports, enable the port to receive the interrupt.

      2) Set conditions such as active state upon initialization.

      3) Clear interrupt requests.

      4) Enable interrupts.

(Note 2) Each settings must be performed while interrupts are disabled.

(Note 3) For clearing the Standby modes , 10 interrupt factors (INT3 to INTF) for TMPM372, 8 interrupt factors (INT4 to INT7, INT9 to INTF) for TMPM373, 7 interrupt factors (INT5 to INT7, INT9 to INTF) for TMPM374 are available. CGIMCGA to CGIMCGD Resisters in CG are used for selecting edge/level of active state and which aforementioned factors are used for clearing the standby modes.

(Note 4) In case the standby mode clear is not required, INT3 to INTF can be used as a normal interrupt without setting CGIMCGA to CGIMCGD Resisters in CG.

8.6.3.5 CG Interrupt Clear Request Register

This resister clear the Interrupt request from INT0 to INTF.

CGICRCG

	7	6	5	4	3	2	1	0
bit Symbol				ICRCG4	ICRCG3	ICRCG2	ICRCG1	ICRCG0
Read/Write		R				W		
After reset		0		0	0	0	0	0
Function	"0" is read.			Clear interrupt requests 0_0000: Reserved 0_1000:INT8 0_0001: Reserved 0_1001: Reserved 0_0010: Reserved 0_1010: Reserved 0_0011:INT3 0_1011: Reserved 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_0000 to 1_1111: setting prohibited * "0" is read.				
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write				R				
After reset				0				
Function				"0" is read.。				
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write								
After reset								
Function				"0" is read.。				
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write				R				
After reset				0				
Function				"0" is read.				

Note 1: TMPM373 doesn't have INT3 and INT8.
Note 2: TMPM374 doesn't have INT3, INT4 and INT8.

8.6.3.6   Reset Flag Register

Reset Flag Resister is a resister for reading internal Reset generation status per generation factors.

Since this register is not cleared automatically, it is necceary to write "0" to clear the register.

CGRSTFLG		7	6	5	4	3	2	1	0
	bit Symbol			OFDRSTF	SYSRSTF	VLTDRSTF	WDTRSTF	PINRSTF	PONRSTF
	Read/Write	R		R/W	R/W	R/W	R/W	R/W	R/W
	After power on reset	0	0	0	0	0	0	0	1 / 0
	Function	"0" is read.		OFD reset flag 0 : "0" is written 1 : Reset from OFD	Debug reset flag 0 : "0" is written 1 : Reset from debugger	VLTD reset flag. 0 : "0" is written 1 : Reset from VLTD	WDT reset flag 0 : "0" is written 1 : Reset from WDT	RESETpin flag 0 : "0" is written 1 : Reset from RESET pin	Power On Reset flag 0 : "0" is written 1 : Reset from Power On Reset
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After power on reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After power on reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After power on reset	0	0	0	0	0	0	0	0
	Function	"0" is read.							

(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 register is not cleared automatically. Write "0" to clear the register.

9 Input/Output Ports

9.1 Port registers

PxDATA **: Port data register**
To read/write port data.

PxCR **: Output control Register**
To control enable/disable output.
* To enable/disable input, controled by PxIE register.

PxFRn **: Function control Register**
To set functions. An assigned function can be activated by setting "1".

PxOD **: Open Drain control register**
To switch CMOS output or open drain output.

PxPUP **: Port Pull-Up control register**
To enable/disable pull-up resister.

PxPDN **: Port Pull-Down control register**
To enable/disable pull-down resister.

PxIE **: Input Enable control register**
To enable/disable input. Default setting is disabled for avoiding through current.

9.2 Port Functions

9.2.1 Port States in Stop Mode

Input and output in Stop mode are enabled/disabled by the CGSTBYCR<DRVE> bit in the Standby Control Register

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 and PxCR are enabled.

The differences are summarized in the table shown below.

Table 9.1 Differences of <DRVE> setting

Port/Except Port	Function	Port	Setting	I/O	<DRVE>=0	<DRVE>=1
Port	X2	PM1	CGOSCCR <HOSCON>=1	Output only	"H" level output	
	X1	PM0		Input only	×	
	TMS TCK TDI TRST	PB3 PB4 PB6 PB7	PBFR<n>=1 PBIE<n>=1	Input	O	
	TDO	PB5	PBFR<5>=1 PBCR<5>=1	Output	Enable when data is valid. Disable when data is invalid.	
	SWCLK	PB4	PBFR<4>=1 PBIE<4>=1	Input	O	
	SWDIO	PB3	PBFR<3>=1 PBCR<3>=1 PBIE<3>=1	Input	O	
				Output	Enable when data is valid. Disable when data is invalid.	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	PB0 PB1 PB2 PB5	PBFR<n>=1 PBCR<n>=1	Output	O	
	INT3 INT4 INT5 INT6 INT7 INT8 INTC INTD INTE INTF	PA0 PA2 PE4 PE6 PE7 PA7 PJ6 PJ7 PK0 PK1	PxFR<n>=1 PxIE<n>=1	Input	O	
	Other function pins other than the above or the ports that are used as general purpose ports.		PxIE<n>=1	Input	×	O
			PxCR<n>=1	Output	×	O
Except Port	RESET ,MODE			Input only	O	O

Note 1: O：Input or Output enabled, ×：Input or Output disabled
Note 2: TMPM373 doesn't have PA0 and PA7 pin. TMPM374 doesn't have PA0, PA2 and PA7 pin.

9.2.2 Port A

Product No.	Exsitence pins
TMPM372FWUG	PA0 to PA7
TMPM373FWDUG	PA2, PA4 to PA6
TMPM374FWUG	PA4 to PA6

The port A is a general-purpose, 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 and the external interrupt input and 16-bit timer input and 16-bit timer output.

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

To use the external interrupt input for releasing STOP mode, select this function in the PAFR2 register and enable input in the PAIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

> **(Note)** In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting as long as input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

Port A circiut types

	7	6	5	4	3	2	1	0
Type	T12	T11	T13	T9	T2	T12	T2	T12

Port A register

		7	6	5	4	3	2	1	0
PADATA	Bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
(0x4000_0000)	Read/Write	R/W							
	After reset	"0"							

Note 1: Always write "0" to PA7, PA3, PA1 and PA0 in TMPM373.

Note 2: Always write "0" to PA7, PA3, PA2, PA1 and PA0 in TMPM374.

Port A control register

		7	6	5	4	3	2	1	0
PACR	Bit Symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
(0x4000_0004)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Output 0: disabled 1: enabled							

Note 1: Always write "0" to PA7C, PA3C, PA1C and PA0C in TMPM373.

Note 2: Always write "0" to PA7C, PA3C, PA2C, PA1C and PA0C in TMPM374.

Port A function register 1

PAFR1 (0x4000_0008)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7F1	PA6F1	PA5F1	PA4F1	PA3F1	PA2F1	PA1F1	PA0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:TB4IN	0:PORT 1:RXD1	0:PORT 1:TXD1	0:PORT 1:SCLK1	0:PORT 1:TB1OUT	0:PORT 1:TB1IN	0:PORT 1:TB0OUT	0:PORT 1:TB0IN

Note 1: Always write "0" to PA7F1, PA3F1, PA1F1 and PA0F1 in TMPM373.

Note 2: Always write "0" to PA7F1, PA3F1, PA2F1, PA1F1 and PA0F1 in TMPM374.

Port A function register 2

PAFR2 (0x4000_000C)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7F2	PA6F2	PA5F2	PA4F2	—	PA2F2	—	PA0F2
	Read/Write	R/W				R	R/W	R	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1 INT8	0:PORT 1:TB6IN	0:PORT 1:TB6OUT	0:PORT 1:$\overline{\text{CTS1}}$	"0" is read.	0:PORT 1:INT4	"0" is read.	0:PORT 1:INT3

Note 1: Always write "0" to PA7F2 and PA0F2 in TMPM373.

Note 2: Always write "0" to PA7F2, PA2F2 and PA0F2 in TMPM374.

Port A open drain Control register

PAOD (0x4000_ 0028)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7OD	PA6OD	PA5OD	PA4OD	PA3OD	PA2OD	PA1OD	PA0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1: open drain							

Note 1: Always write "0" to PA7OD, PA3OD, PA1OD and PA0OD in TMPM373.

Note 2: Always write "0" to PA7OD, PA3OD, PA2OD, PA1OD and PA0OD in TMPM374.

Port A pull-up control register

PAPUP (0x4000_002C)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7UP	PA6UP	PA5UP	PA4UP	PA3UP	PA2UP	PA1UP	PA0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: disabled 1: enabled							

Note 1: Always write "0" to PA7UP, PA3UP, PA1UP and PA0UP in TMPM373.

Note 2: Always write "0" to PA7UP, PA3UP, PA2UP, PA1UP and PA0UP in TMPM374.

Port A pull-down control register

	7	6	5	4	3	2	1	0
Bit Symbol	PA7DN	PA6DN	PA5DN	PA4DN	PA3DN	PA2DN	PA1DN	PA0DN
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-down 0: disabled 1: enabled							

PAPDN
(0x4000_0030)

Note 1: Always write "0" to PA7DN, PA3DN, PA1DN and PA0DN in TMPM373.

Note 2: Always write "0" to PA7DN, PA3DN, PA2DN, PA1DN and PA0DN in TMPM374.

Port A input enable control register

	7	6	5	4	3	2	1	0
Bit Symbol	PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: enabled							

PAIE
(0x4000_0038)

Note 1: Always write "0" to PA7IE, PA3IE, PA1IE and PA0IE in TMPM373.

Note 2: Always write "0" to PA7IE, PA3IE, PA2IE, PA1IE and PA0IE in TMPM374.

9.2.3　Port B

Product No.	Exsitence pins
TMPM372FWUG	PB0 to PB7
TMPM373FWDUG	PB3 to PB6
TMPM374FWUG	PB3 to PB6

The port B is a general-purpose, 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.

Reset initializes PB3, PB4, PB5, PB6 and PB7 to perform debug communication function.When PB3 functions as the TMS or SWDIO, input, output and pull-up are enabled. When PB4 functions as the TCK or SWCLK, input, pull-down are enabled. When PB5 functions as the TDO or SWV, output is enabled. When PB6 functions TDI, input, pull-up are enabled. 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 PB0,PB1,PB2,PB3 and PB5 are configured to alternated function port for debug function, outputs are enabled even during Stop mode regardless of the CGSTBYCR<DRVE> bit setting.**

(Note 6)　**TMPM373/374 don't have PB0, PB1, PB2 and PB7 pin.**

Port B circiut types

	7	6	5	4	3	2	1	0
Type	T7	T7	T19	T8	T6	T18	T18	T18

Port B register

		7	6	5	4	3	2	1	0
PBDATA	Bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
(0x4000_0040)	Read/Write	R/W							
	After reset	"0"							

Note: Always write "0" to PB7, PB2, PB1 and PB0 in TMPM373/374.

Port B control register

		7	6	5	4	3	2	1	0
PBCR	Bit Symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
(0x4000_0044)	Read/Write	R/W							
	After reset	0	0	1	0	1	0	0	0
	Function	Output 0: disabled 1: enabled							

Note: Always write "0" to PB7C, PB2C, PB1C and PB0C in TMPM373/374.

Port B function register 1

PBFR1		7	6	5	4	3	2	1	0
(0x4000_0048)	Bit Symbol	PB7F1	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
	Read/Write	R/W							
	After reset	1	1	1	1	1	0	0	0
	Function	0:PORT 1:$\overline{\text{TRST}}$	0:PORT 1:TDI	0:PORT 1:TDO/ SWV	0:PORT 1:TCK/ SWCLK	0:PORT 1:TMS/ SWDIO	0:PORT 1:TRACE DATA1	0:PORT 1:TRACE DATA0	0:PORT 1:TRACE CLK

Note: Always write "1" to PB7F1 and write "0" to PB2F1, PB1F1 and PB0F1 in TMPM373/374.

Port B open drain Control register

PBOD		7	6	5	4	3	2	1	0
(0x4000_ 0068)	Bit Symbol	PB7OD	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1: open drain							

Note : Always write "0" to PB7OD, PB2OD, PB1OD and PB0OD in TMPM373/374

Port B pull-up control register

PBPUP		7	6	5	4	3	2	1	0
(0x4000_006C)	Bit Symbol	PB7UP	PB6UP	PB5UP	PB4UP	PB3UP	PB2UP	PB1UP	PB0UP
	Read/Write	R/W							
	After reset	1	1	0	0	1	0	0	0
	Function	Pull-up 0: disabled 1: enabled							

Note: Always write "1" to PB7UP and write "0" to PB2UP, PB1UP and PB0UP in TMPM373/374.

Port B pull-down control register

PBPDN		7	6	5	4	3	2	1	0
(0x4000_0070)	Bit Symbol	PB7DN	PB6DN	PB5DN	PB4DN	PB3DN	PB2DN	PB1DN	PB0DN
	Read/Write	R/W							
	After reset	0	0	0	1	0	0	0	0
	Function	Pull-down 0: disabled 1: enabled							

Note: Always write "0" to PB7DN, PB2DN, PB1DN and PB0DN in TMPM373/374.

Port B input enable control register

PBIE		7	6	5	4	3	2	1	0
(0x4000_0078)	Bit Symbol	PB7IE	PB6IE	PB5IE	PB4IE	PB3IE	PB2IE	PB1IE	PB0IE
	Read/Write	R/W							
	After reset	1	1	0	1	1	0	0	0
	Function	Input 0: disabled 1: enabled							

Note: Always write "1" to PB7IE and write "0" to PB2IE, PB1IE and PB0IE in TMPM373/374.

9.2.4 Port D

Product No.	Exsitence pins
TMPM372FWUG	PD4 to PA6
TMPM373FWDUG	PD6
TMPM374FWUG	Nothing

The port D is a general-purpose, 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, 16-bit timer input/output, the external interrupt input and encoder input.

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

To use the external interrupt input for releasing STOP mode, select this function in the PDFR1 register and enable input in the PDIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

(Note)	**TMPM374 doesn't have Pord D.**

Port D circuit types

	7	6	5	4	3	2	1	0
Type	−	T3	T2	T9	−	−	−	−

Port D register

		7	6	5	4	3	2	1	0
PDDATA	Bit Symbol	-	PD6	PD5	PD4	-	-	-	-
(0x4000_00C0)	Read/Write	R	R/W			R			
	After reset	"0"							

Note 1: Always write "0" to PD5 and PD4 in TMPM373.

Note 2: Do not access to PDDATA in TMPM374.

Port D control register

		7	6	5	4	3	2	1	0
PDCR	Bit Symbol	-	PD6C	PD5C	PD4C	-	-	-	-
(0x4000_00C4)	Read/Write	R	R/W			R			
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	Output 0: disabled 1: enabled			"0" is read.			

Note 1: Always write "0" to PD5C and PD4C in TMPM373.

Note 2: Do not access to PDCR in TMPM374.

Port D function register 1

	7	6	5	4	3	2	1	0
Bit Symbol	-	PD6F1	PD5F1	PD4F1	-	-	-	-
Read/Write	R	R/W			R			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	0:PORT 1:RXD2	0:PORT 1:TXD2	0:PORT 1:SCLK2	"0" is read.			

PDFR1
(0x4000_00C8)

Note 1: Always write "0" to PD5F1 and PD4F1 in TMPM373.

Note 2: Do not access to PDFR1 in TMPM374.

Port D function register 2

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	PD4F2	-	-	-	-
Read/Write	R			R/W	R			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.			0:PORT 1:$\overline{CTS2}$	"0" is read.			

PDFR2
(0x4000_00CC)

Note 1: Always write "0" to PD4F2 in TMPM373.

Note 2: Do not access to PDFR2 in TMPM374.

Port D   open drain control register

	7	6	5	4	3	2	1	0
Bit Symbol	-	PD6OD	PD5OD	PD4OD	-	-	-	-
Read/Write	R	R/W			R			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	0:CMOS 1: open drain			"0" is read.			

PDOD
(0x4000_ 00E8)

Note 1: Always write "0" to PD5OD and PD4OD in TMPM373.

Note 2: Do not access to PDOD in TMPM374.

Port D pull-up control register

	7	6	5	4	3	2	1	0
Bit Symbol	-	PD6UP	PD5UP	PD4UP	-	-	-	-
Read/Write	R	R/W			R			
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Pull-up 0:disabled 1:enabled			"0" is read.			

PDPUP
(0x4000_00EC)

Note 1: Always write "0" to PD5UP and PD4UP in TMPM373.

Note 2: Do not access to PDPUP in TMPM374.

Port D pull-down control register

PDPDN		7	6	5	4	3	2	1	0
(0x4000_00F0)	Bit Symbol	-	PD6DN	PD5DN	PD4DN	-	-	-	-
	Read/Write	R	R/W			R			
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	Pull-down 0:disabled 1:enabled			"0" is read.			

Note 1: Always write "0" to PD5DN and PD4DN in TMPM373.

Note 2: Do not access to PDPDN in TMPM374.

Port D input enable control register

PDIE		7	6	5	4	3	2	1	0
(0x4000_00F8)	Bit Symbol	-	PD6IE	PD5IE	PD4IE	-	-	-	-
	Read/Write	R	R/W			R			
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read.	Input 0:disabled 1:enabled			"0" is read.			

Note 1: Always write "0" to PD5IE and PD4IE in TMPM373.

Note 2: Do not access to PDIE in TMPM374.

9.2.5 Port E

Product No.	Exsitence pins
TMPM372FWUG	PE0 to PE7
TMPM373FWDUG	PA0 to PE2, PE4, PE6, PA7
TMPM374FWUG	PA0 to PE2, PE4, PE6, PA7

The port E is a general-purpose, 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, timer input/output and the external interrupt input.

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

To use the external interrupt input for releasing STOP mode, select this function in the PEFR2 register and enable input in the PEIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

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

Port E circuit type

	7	6	5	4	3	2	1	0
Type	T14	T12	T2	T12	T2	T9	T3	T2

Port E register

PEDATA (0x4000_0100)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	Read/Write	R/W							
	After reset	"0"							

Note: Always write "0" to PE5 and PE3 in TMPM373/374.

Port E control register

PECR (0x4000_0104)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Output 0: disabled 1: enabled							

Note: Always write "0" to PE5C and PE3C in TMPM373/374.

Port E functions register 1

PEFR1 (0x4000_0108)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7F1	PE6F1	PE5F1	PE4F1	PE3F1	PE2F1	PE1F1	PE0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:TB3OUT.	0:PORT 1:TB3IN	0:PORT 1:TB2OUT	0:PORT 1: TB2IN	0:PORT 1:TB4OUT	0:PORT 1:SCLK0	0:PORT 1:RXD0	0:PORT 1: TXD0

Note: Always write "0" to PE5F1 and PE3F1 in TMPM373/374.

Port E functions register 2

PEFR2 (0x4000_010C)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7F2	PE6F2	─	PE4F2	─	PE2F2	─	─
	Read/Write	R/W	R/W	R	R/W	R	R/W	R	
	After reset	0	0	0	0	0	0	0	
	Function	0:PORT 1:INT7.	0:PORT 1:INT6	"0" is read.	0:PORT 1:INT5	"0" is read.	0:PORT 1:$\overline{\text{CTS0}}$	"0" is read.	

Port E open drain control register

PEOD (0x4000_ 0128)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7OD	PE6OD	PE5OD	PE4OD	PE3OD	PE2OD	PE1OD	PE0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:Open drain							

Note: Always write "0" to PE5OD and PE3OD in TMPM373/374.

Port E pull-up control register

PEPUP (0x4000_012C)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7UP	PE6UP	PE5UP	PE4UP	PE3UP	PE2UP	PE1UP	PE0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:disabled 1:enabled							

Note: Always write "0" to PE5UP and PE3UP in TMPM373/374.

Port E pull-down control register

PEPDN (0x4000_0130)		7	6	5	4	3	2	1	0
	Bit Symbol	PE7DN	PE6DN	PE5DN	PE4DN	PE3DN	PE2DN	PE1DN	PE0DN
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-down 0:disabled 1:enabled							

Note: Always write "0" to PE5DN and PE3DN in TMPM373/374.

Port E input enable control register

PEIE
(0x4000_0138)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7IE	PE6IE	PE5IE	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0:disabled 1:enabled							

Note: Always write "0" to PE5IE and PE3IE in TMPM373/374.

9.2.6 Port F

Product No.	Exsitence pins
TMPM372FWUG	PF0 to PF4
TMPM373FWDUG	PF0 to PF4
TMPM374FWUG	PF0, PF2 to PF4

The port F is a general-purpose, 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 functions of the serial interface, timer output and the operation mode setting.

While a reset signal is in "0"state, the PF0 input and pull-up are enabled. At the rising edge of the reset signal, if PF0 is "1", the device enters single mode and boots from the on-chip flash memory. If PF0 is "0", the device enters single boot mode and boots from the internal boot program. For details of single boot mode, refer to Chapter "Flash Memory Operation". After the reset state, the PF0 input and pull-up become disable according to the initial value of PFIE and PFPUP.

All bits of port F except PF0, reset initializes them as general-purpose ports with input, output and pull-down disabled.

Port F circuit type

	7	6	5	4	3	2	1	0
Type	−	−	−	T11	T10	T15	T2	T20

Port F register

		7	6	5	4	3	2	1	0
PFDATA	Bit Symbol	−	−	−	PF4	PF3	PF2	PF1	PF0
(0x4000_0140)	Read/Write	R			R/W				
	After reset	"0" is read.			"0"				

Note: Always write "0" to PF1 in TMPM374.

Port F control register

		7	6	5	4	3	2	1	0
PFCR	Bit Symbol	−	−	−	PF4C	PF3C	PF2C	PF1C	PF0C
(0x4000_0144)	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read.			Output 0: disabled 1: enabled				

Note: Always write "0" to PF1C in TMPM374.

Port F function register 1

		7	6	5	4	3	2	1	0
PFFR1	Bit Symbol	−	−	−	PF4F1	PF3F1	PF2F1	PF1F1	PF0F1
(0x4000_0148)	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read.			0:PORT 1: ENCZ1	0:PORT 1: ENCB1	0:PORT 1: ENCA1	0:PORT 1:TB7OUT	0:PORT 1: TB7IN

Note: Always write "0" to PF1F1 in TMPM374.

Port F function register 2

PFFR2		7	6	5	4	3	2	1	0
(0x4000_014C)	Bit Symbol	−	−	−	PF4F2	PF3F2	PF2F2	−	−
	Read/Write	R			R/W			R	
	After reset	0			0	0	0	0	
	Function	"0" is read.			0:PORT 1:RXD3	0:PORT 1:TXD3	0:PORT 1:SCLK3	"0" is read.	

Port F function register 3

PFFR3		7	6	5	4	3	2	1	0
(0x4000_0150)	Bit Symbol	−	−	−	−	−	PF2F3	−	−
	Read/Write	R					R/W	R	
	After reset	0					0	0	
	Function	"0" is read.					0:PORT 1:CTS3	"0" is read.	

Port F open drain control register

PFOD		7	6	5	4	3	2	1	0
(0x4000_0168)	Bit Symbol	−	−	−	PF4OD	PF3OD	PF2OD	PF1OD	PF0OD
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read			0:CMOS 1:Open drain				

Note: Always write "0" to PF1OD in TMPM374.

Port F pull-up control register

PFPUP		7	6	5	4	3	2	1	0
(0x4000_016C)	Bit Symbol	−	−	−	PF4UP	PF3UP	PF2UP	PF1UP	PF0UP
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read			Pull-up 0:disabled 1:enabled				

Note: Always write "0" to PF1UP in TMPM374.

Port F pull-down control register

PFPDN		7	6	5	4	3	2	1	0
(0x4000_0170)	Bit Symbol	−	−	−	PF4DN	PF3DN	PF2DN	PF1DN	PF0DN
	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read			Pull-down 0:disabled 1:enabled				

Note: Always write "0" to PF1DN in TMPM374.

Port F input enable control register

		7	6	5	4	3	2	1	0
PFIE	Bit Symbol	—	—	—	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
(0x4000_0178)	Read/Write	R			R/W				
	After reset	0			0	0	0	0	0
	Function	"0" is read			Input 0:disabled 1:enabled				

Note: Always write "0" to PF1IE in TMPM374.

9.2.7 Port G

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 three-phase motor control (PMD).

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

Port G circuit type

	7	6	5	4	3	2	1	0
Type	T3	T3	T1	T1	T1	T1	T1	T1

Port G register

		7	6	5	4	3	2	1	0
PGDATA	Bit Symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
(0x4000_0180)	Read/Write				R/W				
	After reset				"0"				

Port G control register

		7	6	5	4	3	2	1	0
PGCR	Bit Symbol	PG7C	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
(0x4000_0184)	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0
	Function				Output 0: disabled 1: enabled				

Port G function register 1

		7	6	5	4	3	2	1	0
PGFR1	Bit Symbol	PG7F1	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
(0x4000_0188)	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1: OVV1	0:PORT 1: EMG1	0:PORT 1: ZO1	0:PORT 1: WO1	0:PORT 1: YO1	0:PORT 1: VO1	0:PORT 1: XO1	0:PORT 1: UO1

Port G open drain control register

		7	6	5	4	3	2	1	0
PGOD	Bit Symbol	PG7OD	PG6OD	PG5OD	PG4OD	PG3OD	PG2OD	PG1OD	PG0OD
(0x4000_01A8)	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0
	Function				0:CMOS 1:Open drain				

Port G pull-up control register

PGPUP		7	6	5	4	3	2	1	0
(0x4000_01AC)	Bit Symbol	PG7UP	PG6UP	PG5UP	PG4UP	PG3UP	PG2UP	PG1UP	PG0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0:disabled 1:enabled							

Port G pull-down control register

PGPDN		7	6	5	4	3	2	1	0
(0x4000_01B0)	Bit Symbol	PG7DN	PG6DN	PG5DN	PG4DN	PG3DN	PG2DN	PG1DN	PG0DN
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-down 0:disabled 1:enabled							

Port G input enable control register

PGIE		7	6	5	4	3	2	1	0
(0x4000_01B8)	Bit Symbol	PG7IE	PG6IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled							

9.2.8 Port I

The port I is a general-purpose, 1-bit input/output port. For this port, input and output can be specified. Besides the general-purpose input/output function, the port I performs the analog input of the A/D 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, port J and port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.**

Port I circuit type

	7	6	5	4	3	2	1	0
Type	—	—	—	—	T16	—	—	—

Port I register

		7	6	5	4	3	2	1	0
PIDATA	Bit Symbol	—	—	—	—	PI3	—	—	—
(0x4000_0200)	Read/Write	R				R/W	R		
	After reset	"0"							

Port I control register

		7	6	5	4	3	2	1	0
PICR	Bit Symbol	—	—	—	—	PI3C	—	—	—
(0x4000_0204)	Read/Write	R				R/W	R		
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read				Output 0: disabled 1: enabled	"0" is read		

Port I open drain control register

		7	6	5	4	3	2	1	0
PIOD	Bit Symbol	—	—	—	—	PI3OD	—	—	—
(0x4000_ 0228)	Read/Write	R				R/W	R		
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read				0: CMOS 1: open drain	"0" is read		

Port I pull-up control register

		7	6	5	4	3	2	1	0
PIPUP	Bit Symbol	—	—	—	—	PI3UP	—	—	—
(0x4000_022C)	Read/Write	R				R/W	R		
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read				Pull-up 0: disabled 1: enabled	"0" is read		

Port I pull-down control register

PIPDN
(0x4000_0230)

	7	6	5	4	3	2	1	0
Bit Symbol	―	―	―	―	PI3DN	―	―	―
Read/Write	R				R/W	R		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read				Pull-down 0: disabled 1: enabled	"0" is read		

Port I input enable control register

PIIE
(0x4000_0238)

	7	6	5	4	3	2	1	0
Bit Symbol	―	―	―	―	PI3IE	―	―	―
Read/Write	R				R/W	R		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read				Input 0: disabled 1: enabled	"0" is read		

9.2.9 Port J

Product No.	Exsitence pins
TMPM372FWUG	PJ0 to PJ7
TMPM373FWDUG	PJ0, PJ5 to PJ7
TMPM374FWUG	PJ0, PJ6, PJ7

The port J is a general-purpose, 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 A/D converter and external interrupt input.

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

To use the external interrupt input for releasing STOP mode, select this function in the PJFR1 register and enable input in the PJIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

(Note 1) In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting as long as input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

(Note 2) Unless you use all the bits of port I, port J and port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

Port J circuit type

	7	6	5	4	3	2	1	0
Type	T17	T17	T16	T16	T16	T16	T16	T16

Port J register

		7	6	5	4	3	2	1	0
PJDATA	Bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
(0x4000_0240)	Read/Write				R/W				
	After reset				"0"				

Note 1: Always write "0" to PJ4, PJ3, PJ2 and PJ1 in TMPM373.

Note 2: Always write "0" to PJ5, PJ4, PJ3, PJ2 and PJ1 in TMPM374.

Port J control register

		7	6	5	4	3	2	1	0
PJCR	Bit Symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
(0x4000_0244)	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0
	Function				Output 0: disabled 1: enabled				

Note 1: Always write "0" to PJ4C, PJ3C, PJ2C and PJ1C in TMPM373.

Note 2: Always write "0" to PJ5C, PJ4C, PJ3C, PJ2C and PJ1C in TMPM374.

Port J function register 1

PJFR1 (0x4000_0248)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7F1	PJ6F1	−	−	−	−	−	−
	Read/Write	R/W		R					
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:INTD	0:PORT 1:INTC	"0" is read					

Port J open drain control register

PJOD (0x4000_ 0268)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7OD	PJ6OD	PJ5OD	PJ4OD	PJ3OD	PJ2OD	PJ1OD	PJ0OD
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:CMOS 1:open drain							

Note 1: Always write "0" to PJ4OD, PJ3OD, PJ2OD and PJ1OD in TMPM373.

Note 2: Always write "0" to PJ5OD, PJ4OD, PJ3OD, PJ2OD and PJ1OD in TMPM374.

Port J pull-up control register

PJPUP (0x4000_026C)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7UP	PJ6UP	PJ5UP	PJ4UP	PJ3UP	PJ2UP	PJ1UP	PJ0UP
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	After reset	Pull-up 0:disabled 1:enabled							

Note 1: Always write "0" to PJ4UP, PJ3UP, PJ2UP and PJ1UP in TMPM373.

Note 2: Always write "0" to PJ5UP, PJ4UP, PJ3UP, PJ2UP and PJ1UP in TMPM374.

Port J pull-down control register

PJPDN (0x4000_0270)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7DN	PJ6DN	PJ5DN	PJ4DN	PJ3DN	PJ2DN	PJ1DN	PJ0DN
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	After reset	Pull-down 0:disabled 1:enabled							

Note 1: Always write "0" to PJ4DN, PJ3DN, PJ2DN and PJ1DN in TMPM373.

Note 2: Always write "0" to PJ5DN, PJ4DN, PJ3DN, PJ2DN and PJ1DN in TMPM374.

Port J input enable control register

PJIE (0x4000_0278)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7IE	PJ6IE	PJ5IE	PJ4IE	PJ3IE	PJ2IE	PJ1IE	PJ0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled							

Note 1: Always write "0" to PJ4IE, PJ3IE, PJ2IE and PJ1IE in TMPM373.

Note 2: Always write "0" to PJ5IE, PJ4IE, PJ3IE, PJ2IE and PJ1IE in TMPM374.

9.2.10 Port K

The port K is a general-purpose, 2-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port K performs the functions as the analog input of the A/D converter and the external interrupt input.

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

To use the external interrupt input for releasing STOP mode, select this function in the PKFR1 register and enable input in the PKIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

(Note 1) **In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting as long as input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.**

(Note 2) **Unless you use all the bits of port I, port J and port K as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.**

Port K circuit type

	7	6	5	4	3	2	1	0
Type	—	—	—	—	—	–	T17	T17

Port K register

		7	6	5	4	3	2	1	0
PKDATA	Bit Symbol	—	—	—	—	—	–	PK1	PK0
(0x4000_0280)	Read/Write	R						R/W	
	After reset	"0" is read						"0"	

Port K control register

		7	6	5	4	3	2	1	0
PKCR	Bit Symbol	—	—	—	—	—	–	PK1C	PK0C
(0x4000_0284)	Read/Write	R						R/W	
	After reset	0						0	0
	Function	"0" is read.						Output 0: disabled 1: enabled	

Port K function register 1

		7	6	5	4	3	2	1	0
PKFR1	Bit Symbol	—	—	—	—	—	–	PK1F1	PK0F1
(0x4000_0288)	Read/Write	R						R/W	
	After reset	0						0	0
	Function	"0" is read						0:PORT 1: INTF	0:PORT 1: INTE

Port K open drain control register

PKOD		7	6	5	4	3	2	1	0
(0x4000_ 02A8)	Bit Symbol	―	―	―	―	―	―	PK1OD	PK0OD
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						0:CMOS 1:open drain	

Port K pull-up control register

PKPUP		7	6	5	4	3	2	1	0
(0x4000_02AC)	Bit Symbol	―	―	―	―	―	―	PK1UP	PK0UP
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Pull-up 0:disabled 1:enabled	

Port K pull-down control register

PKPDN		7	6	5	4	3	2	1	0
(0x4000_02B0)	Bit Symbol	―	―	―	―	―	―	PK1DN	PK0DN
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Pull-down 0:disabled 1:enabled	

Port K input enable control register

PKIE		7	6	5	4	3	2	1	0
(0x4000_02B8)	Bit Symbol	―	―	―	―	―	―	PK1IE	PK0IE
	Read/Write	R						R/W	
	After reset	0						0	0
	Function	"0" is read						Input 0:disabled 1:enabled	

9.2.11 Port M

The port M is a general-purpose, 2-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port M performs the high-speed oscillator1(X1 and X2) by cgosccr<hoscon>=1.

While it become CGOSCCR<HOSCON>=1, each register of PORT_M can not change to write. The procedure when I use it as an outside high-speed oscillator connection terminal look at a chapter of the 'system clock'.(Note1)

Reset initializes all bits of the port M as general-purpose ports with input, output, pull pull-down and high-speed oscillator1 disabled.(Note2)

(Note1) If one of the Port M registers except PMDATA and PMOD is not equal to the initial value, CGOSCCR<HOSCON> can not be set to "1".

(Note2) The high-speed clock chosen after reset cancellation is a built-in high-speed clock.

Therefore, in the initial state, it become PORT_M.

Port M circuit type

	7	6	5	4	3	2	1	0
Type	─	─	─	─	─	─	T21	T21

Port M register

PMDATA (0x4000_0300)		7	6	5	4	3	2	1	0
	Bit Symbol	─	─	─	─	─	─	PM1	PM0
	Read/Write	R						R/W	
	After reset	"0" is read						"0"	

Port M control register

PMCR (0x4000_0304)		7	6	5	4	3	2	1	0
	Bit Symbol	─	─	─	─	─	─	PM1C	PM0C
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Output 0: disabled 1: enabled	

Port Mopen drain control register

PMOD (0x4000_ 0328)		7	6	5	4	3	2	1	0
	Bit Symbol	─	─	─	─	─	─	PM1OD	PM0OD
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						0:CMOS 1:open drain	

Port M pull-up control register

PMPUP		7	6	5	4	3	2	1	0
(0x4000_032C)	Bit Symbol	—	—	—	—	—	—	PM1UP	PM0UP
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Pull-up 0: disabled 1: enabled	

PortM pull-down control register

PMPDN		7	6	5	4	3	2	1	0
(0x4000_0330)	Bit Symbol	—	—	—	—	—	—	PM1DN	PM0DN
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Pull-down 0:disabled 1:enabled	

Port M input enable control register

PMIE		7	6	5	4	3	2	1	0
(0x4000_0338)	Bit Symbol	—	—	—	—	—	—	PM1IE	PM0IE
	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Input 0:disabled 1:enabled	

9.3 Block Diagrams of Ports

9.3.1 Port Types

The ports are classified into 18 types shown below. Please refer to the following pages for the block diagrams of each port type.

Table 9.2 Port type

Type	Port	Function1	Function2	Function3	Analog	Pull-up	Pull-down	OD	Note
T1	i/o	o	-	-	-	R	R	○	Function output triggered by enable signal
T2	i/o	o	-	-	-	R	R	○	
T3	i/o	i	-	-	-	R	R	○	
T4	i/o	i(int)	-	-	-	R	R	○	
T5	i	i(int)	-	-	-	-	-	-	
T6	i/o	i/o	-	-	-	NoR	NoR	○	
T7	i/o	i	-	-	-	NoR	NoR	○	
T8	i/o	i	-	-	-	NoR	NoR	○	
T9	i/o	i/o	i	-	-	R	R	○	
T10	i/o	i	o	-	-	R	R	○	
T11	i/o	i	i	-	-	R	R	○	
T12	i/o	i	i(int)	-	-	R	R	○	
T13	i/o	o	o	-	-	R	R	○	
T14	i/o	o	i(int)	-	-	R	R	○	
T15	i/o	i	i/o	i	-	R	R	○	
T16	i/o	-	-	-	○	R	R	○	
T17	i/o	i(int)	-	-	○	R	R	○	
T18	i/o	o	-	-	-	R	R	○	
T19	i/o	o	-	-	-	NoR	NoR	○	Function output triggered by enable signal
T20	i/o	i	-	-	-	EnR	NoR	○	BOOT input enabled during reset
T21	i/o	-	-	-	-	R	R	○	High-speed osscillator1 (External)

R   : Forced disable during reset.
EnR   : Forced enable during reset.
NoR   : Unaffected by reset.

9.3.2 Type T1

Type T1 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data as well.

The function output is controlled by an enable signal. If enabled, the function data is output.
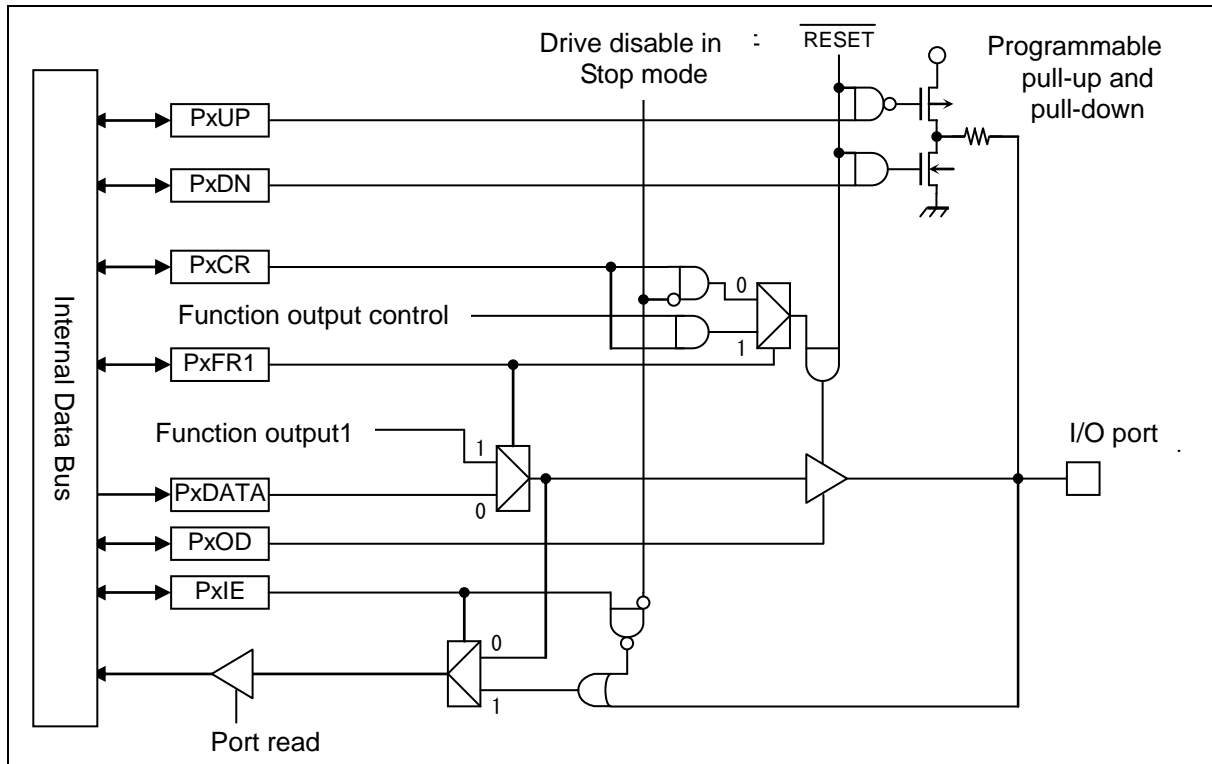
Fig 9-1 Type T1

Input/Output Ports

9.3.3   Type T2

Type T2 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data as well.
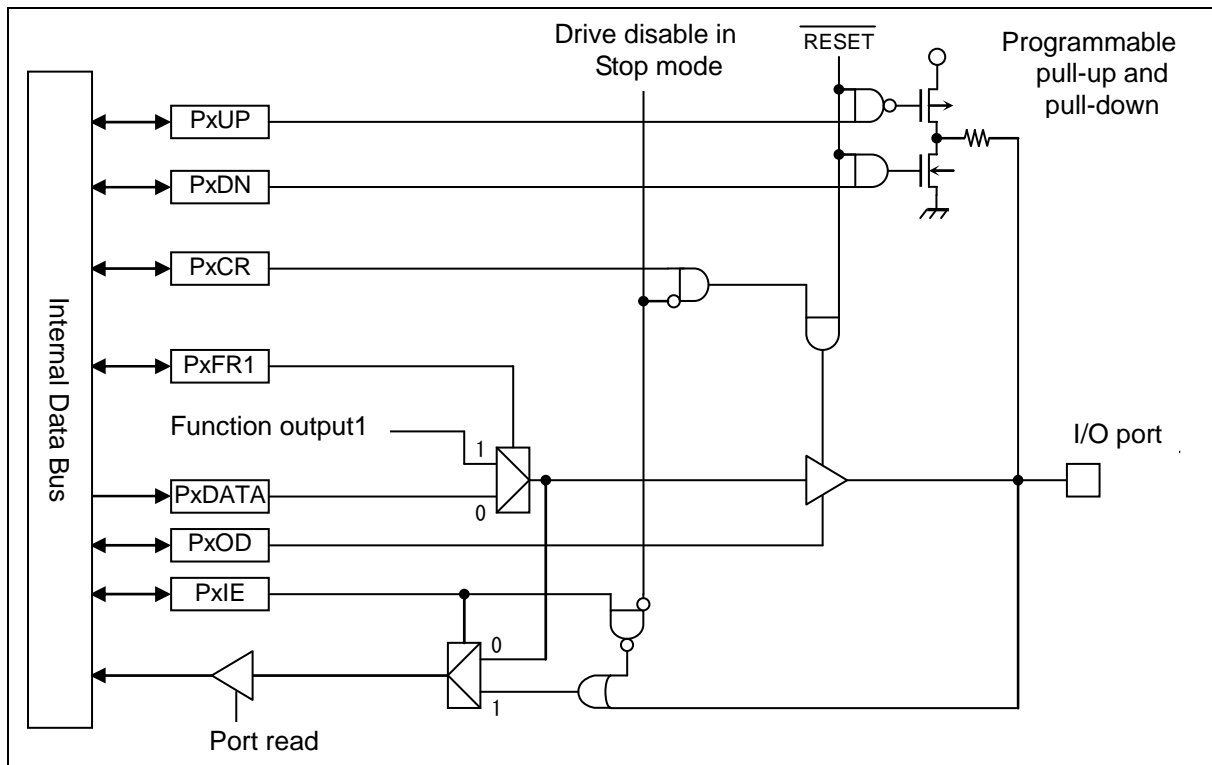
Fig 9-2 Type T2

9.3.4 Type T3

Type T3 is a general-purpose input/output port with pull-up and pull-down. It is used to input function data as well.
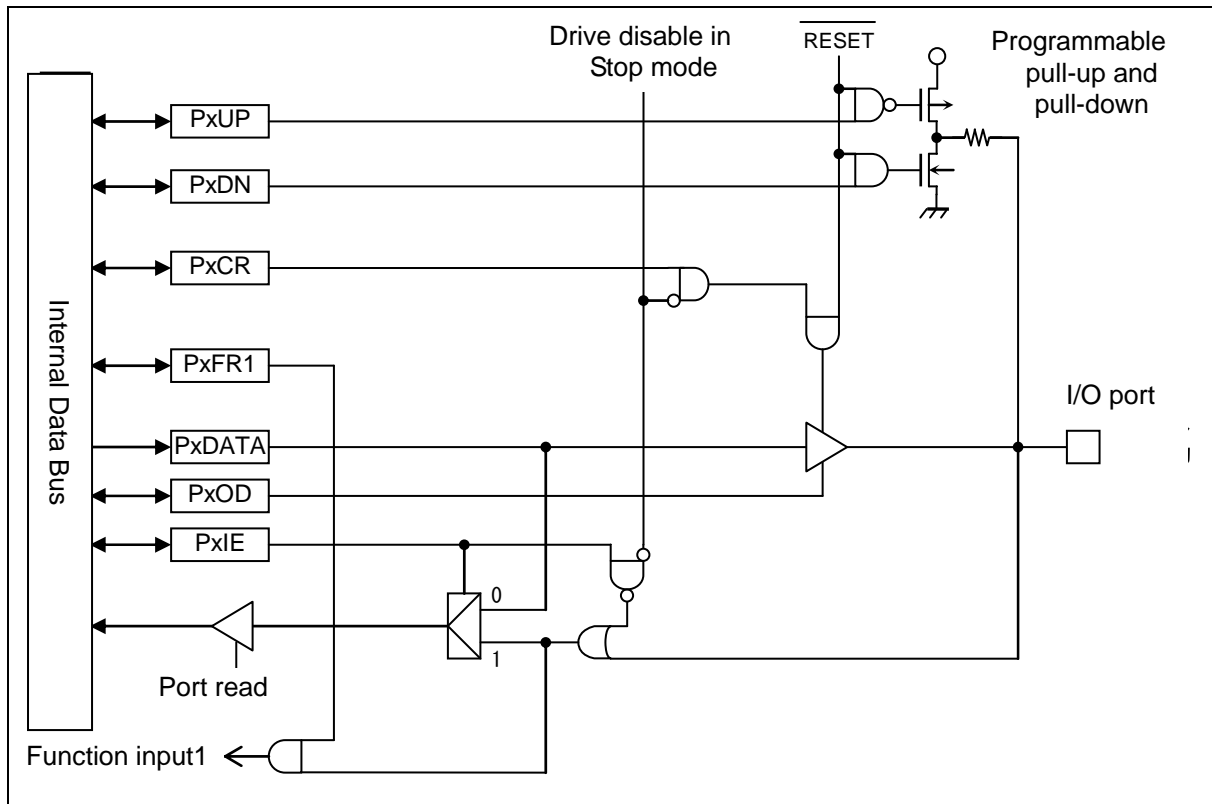
Fig 9-3 Type T3

9.3.5   Type T4

Type T4 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data and interrupt input as well.
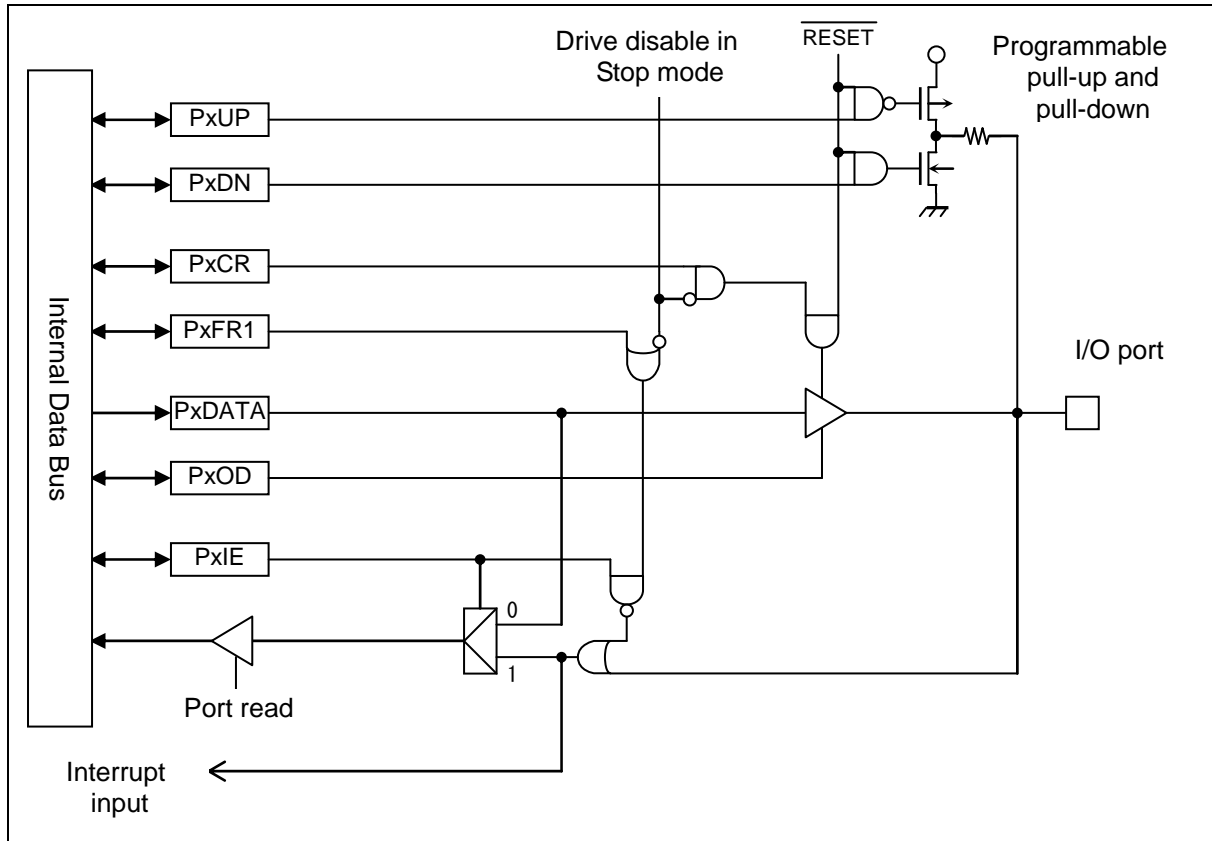
Fig 9.4 Type T4

9.3.6   Type T5

Type T5 is a general-purpose input/output port. It is used to interrupt input as well.

Fig 9.5 Type T5

Input/Output Ports
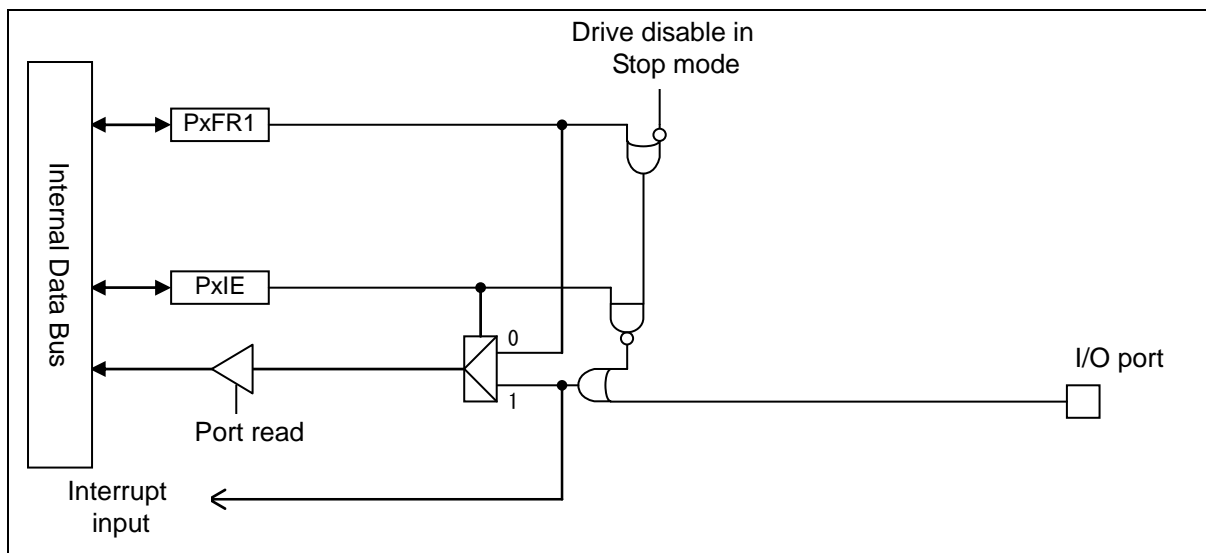
9.3.7   Type T6

Type T6 is a general-purpose port with pull-up and pull-down. It is used to output function data and input interrupt as well.

The function output is controlled by an enable signal. If enabled, the function data is output.
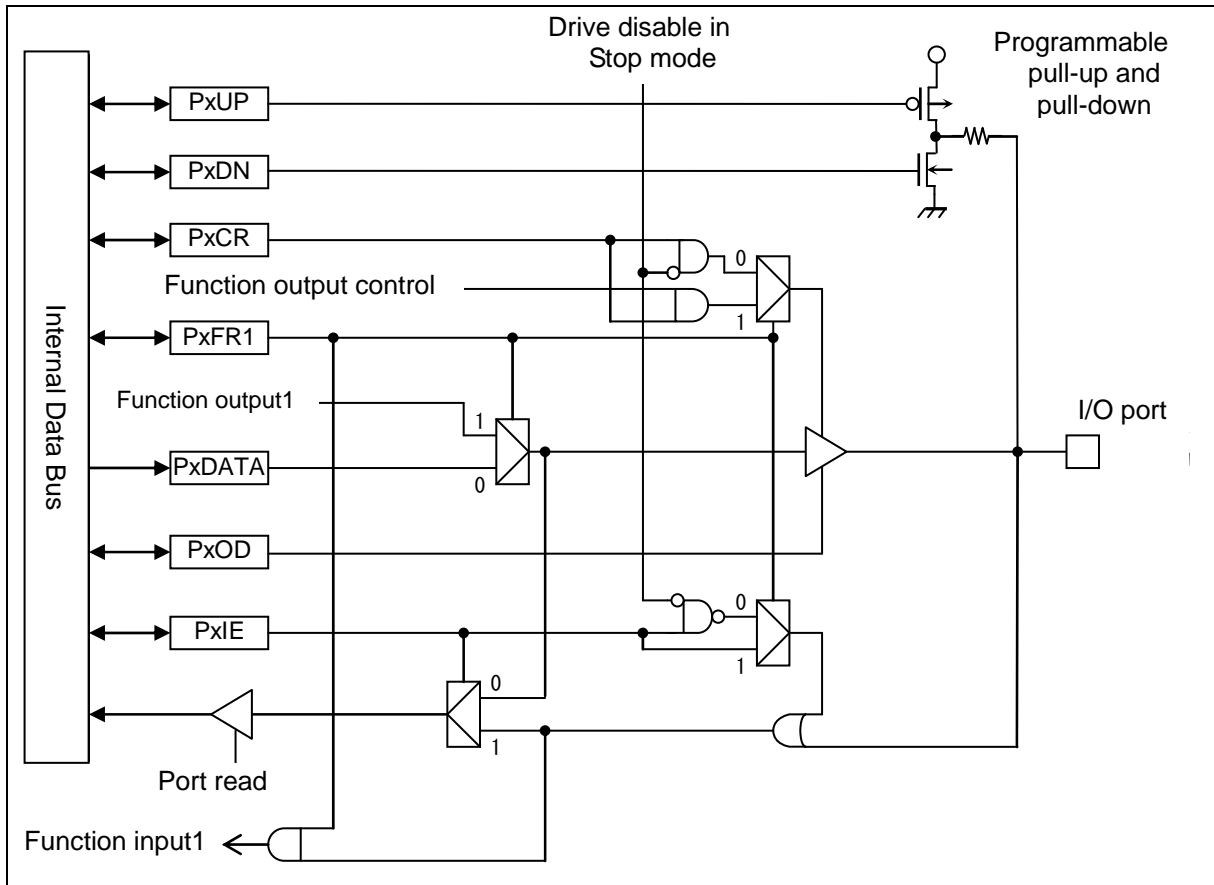
Fig 9-6 Type T6

9.3.8 Type T7

Type T7 is a general-purpose input/output port with pull-up and pull-down. It is used to input function data as well.
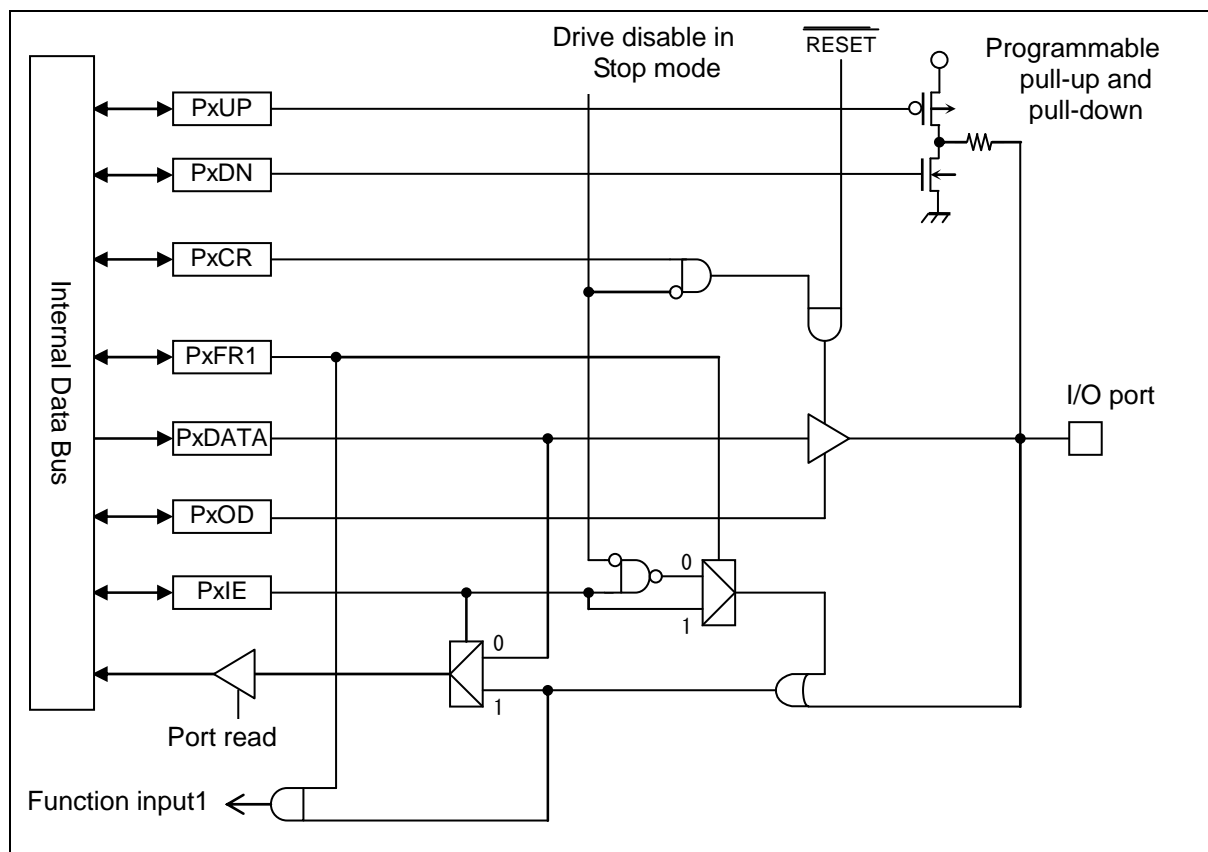
Fig 9-7 Type T7

9.3.9 Type T8

Type T8 is a general-purpose input/output port with pull-up and pull-down. It is used to input function data as well.
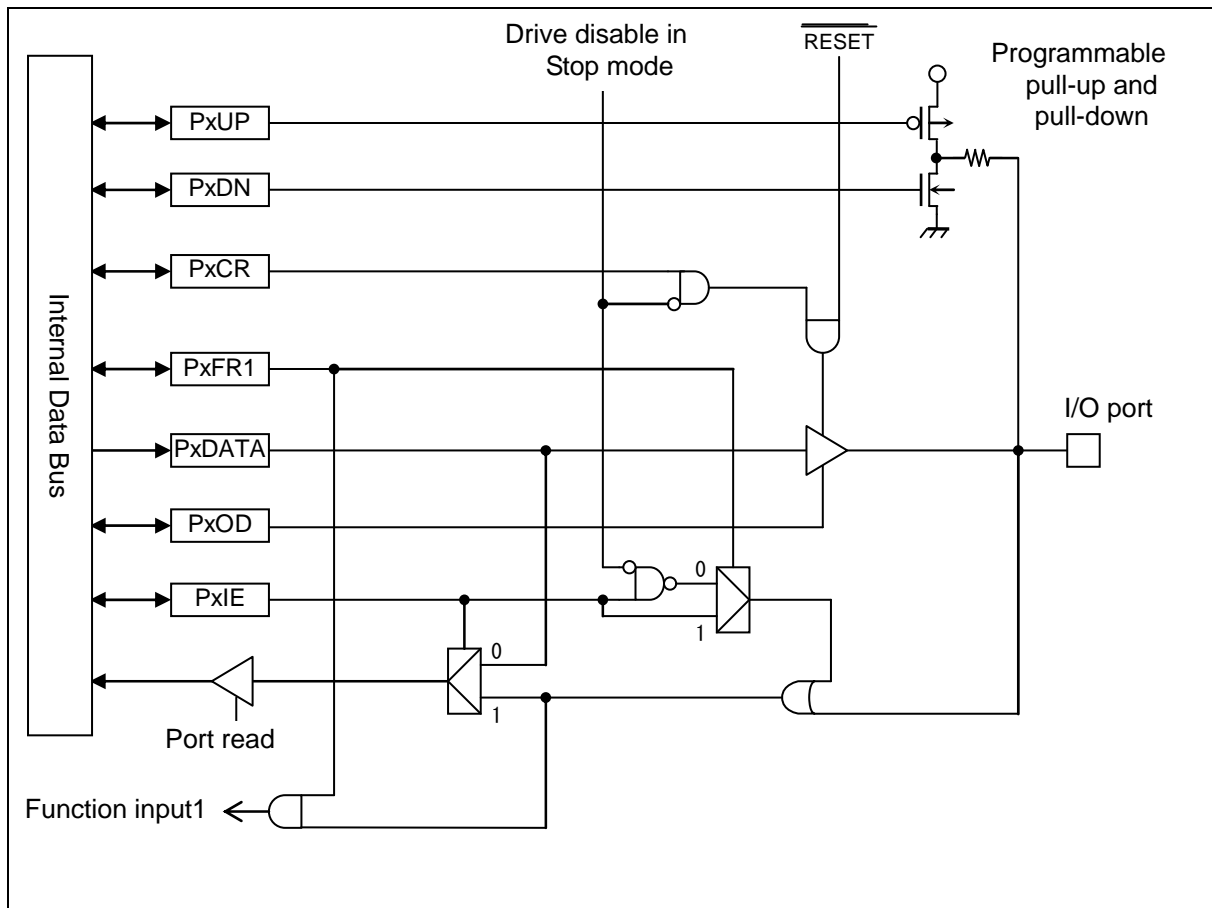
Fig 9-8 Type T8

9.3.10 Type T9

Type T9 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data and two input function data as well.
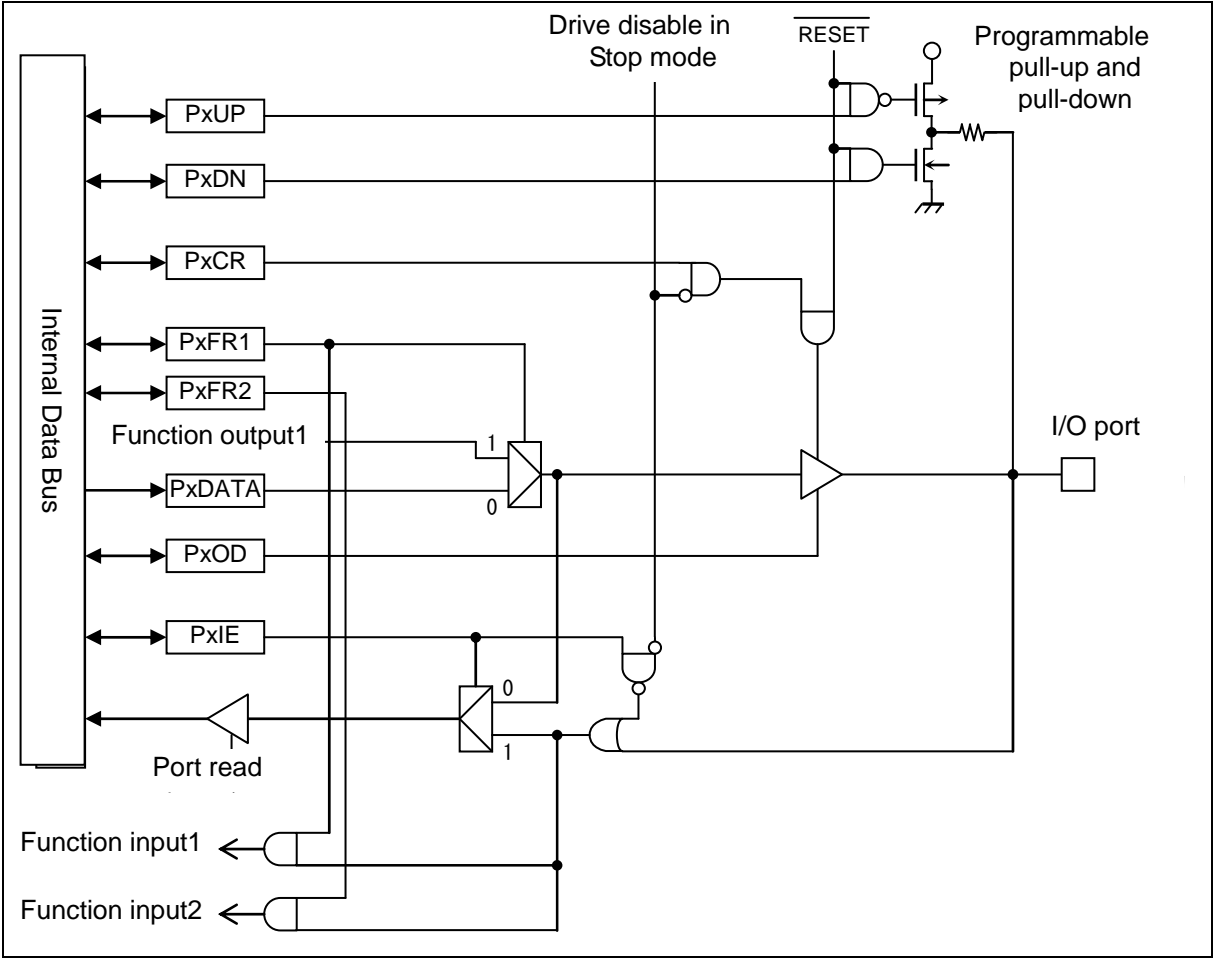
Fig 9-9 Type T9

Input/Output Ports

9.3.11 Type T10

Type T10 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data and input function data as well.
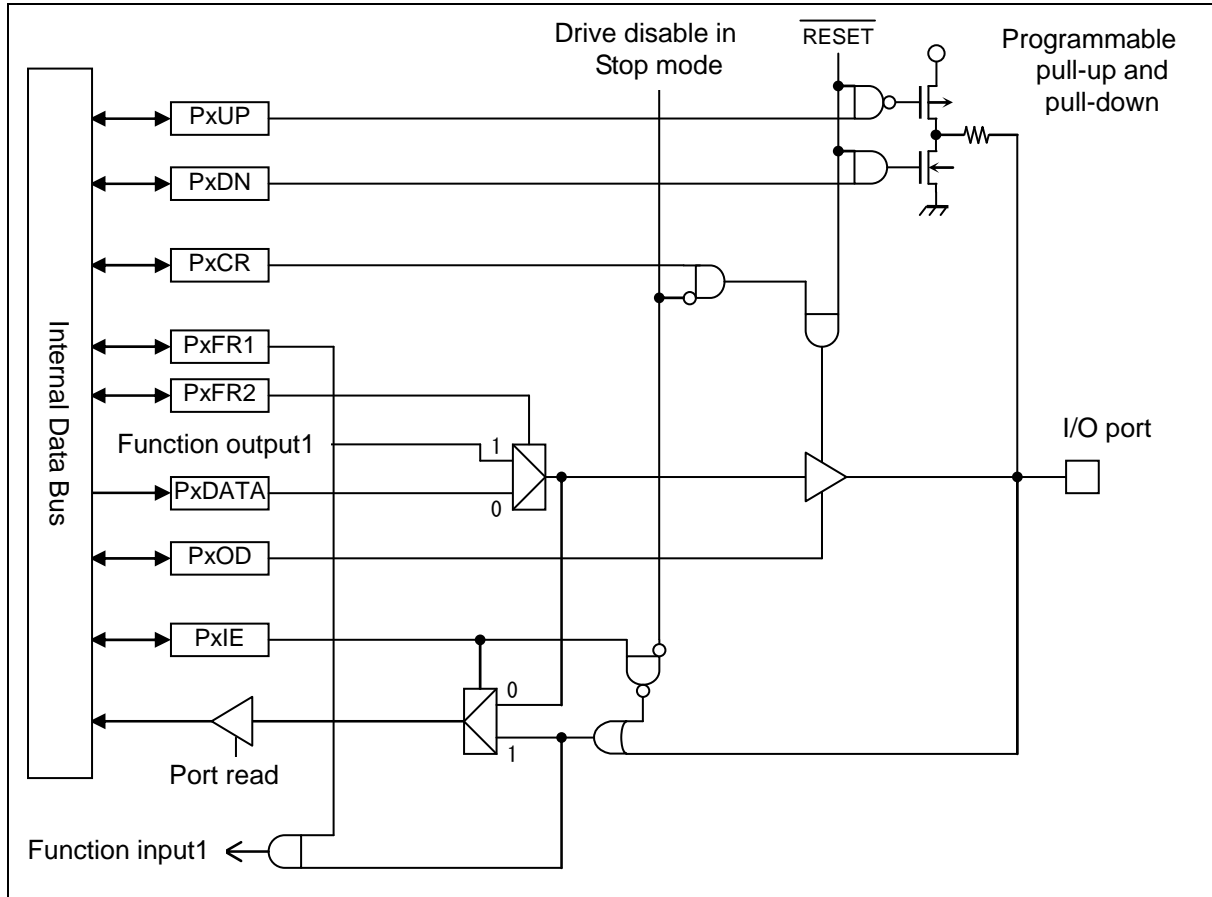
Fig 9-10 Type T10

9.3.12 Type T11

Type T11 is a general-purpose input/output port with pull-up and pull-down. It is used to two input function data as well.
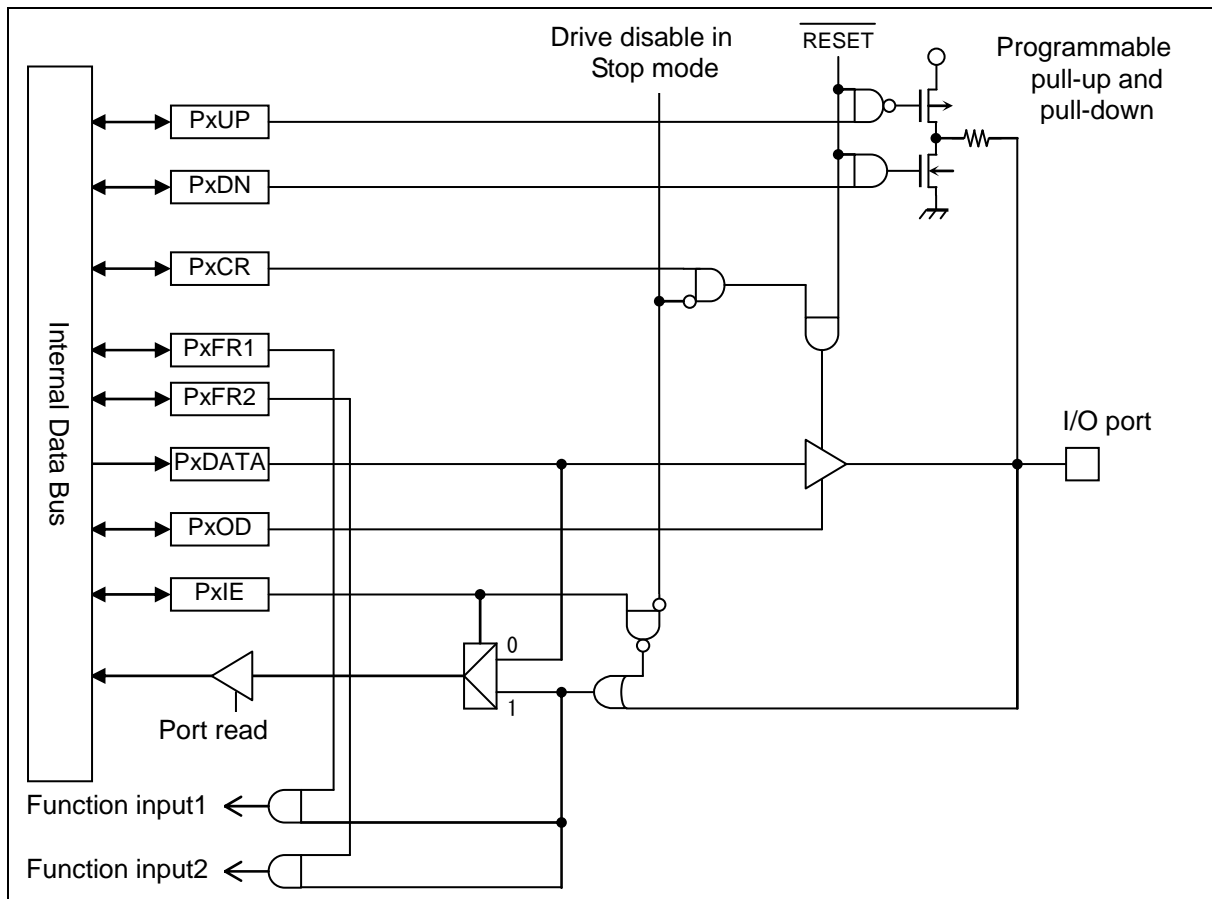
Fig 9-11 Type T11

9.3.13 Type T12

Type T12 is a general-purpose input/output port with pull-up and pull-down. It is used to input function data and interrupt input as well.
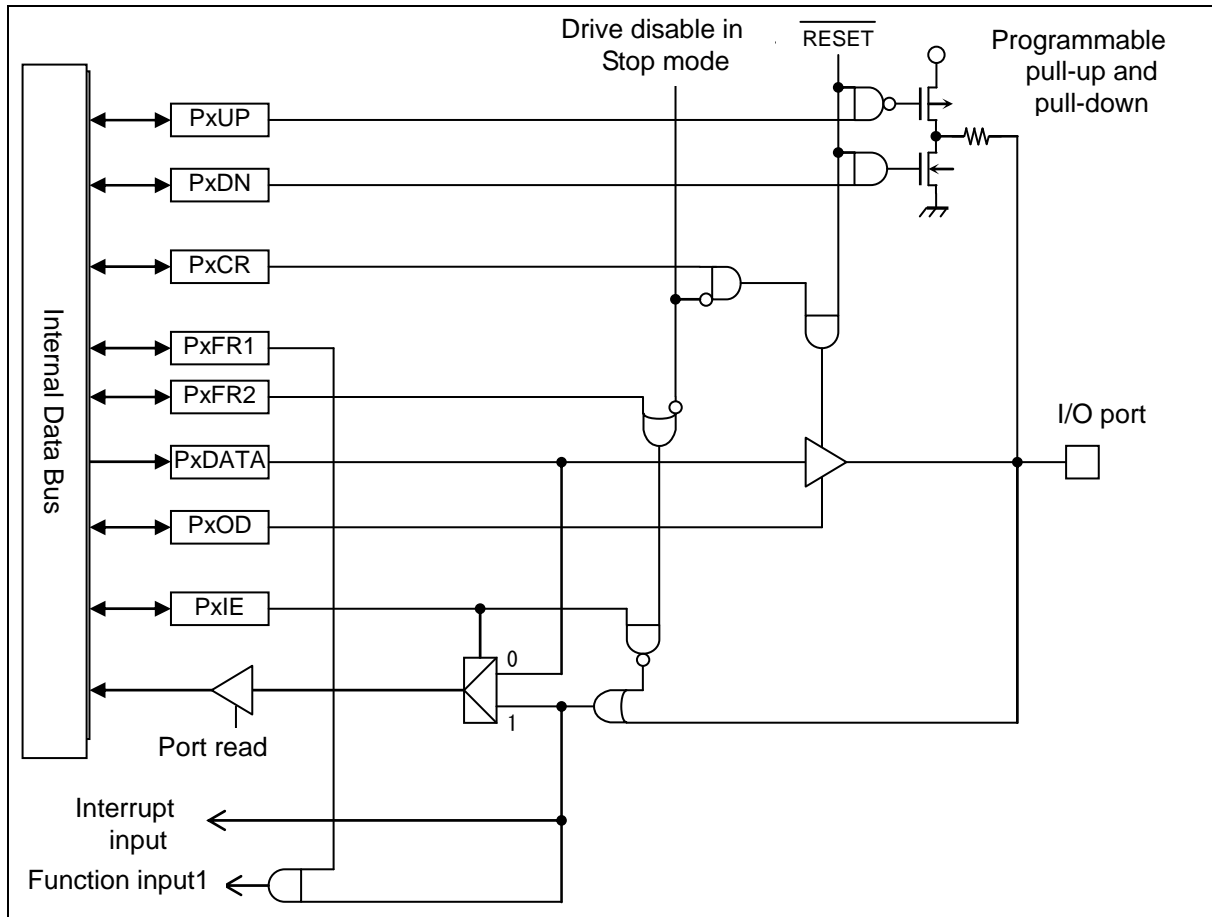
Fig 9-12 Type T12

9.3.14 Type T13

Type T13 is a general-purpose input/output port with pull-up and pull-down. It is used to two output function data as well.
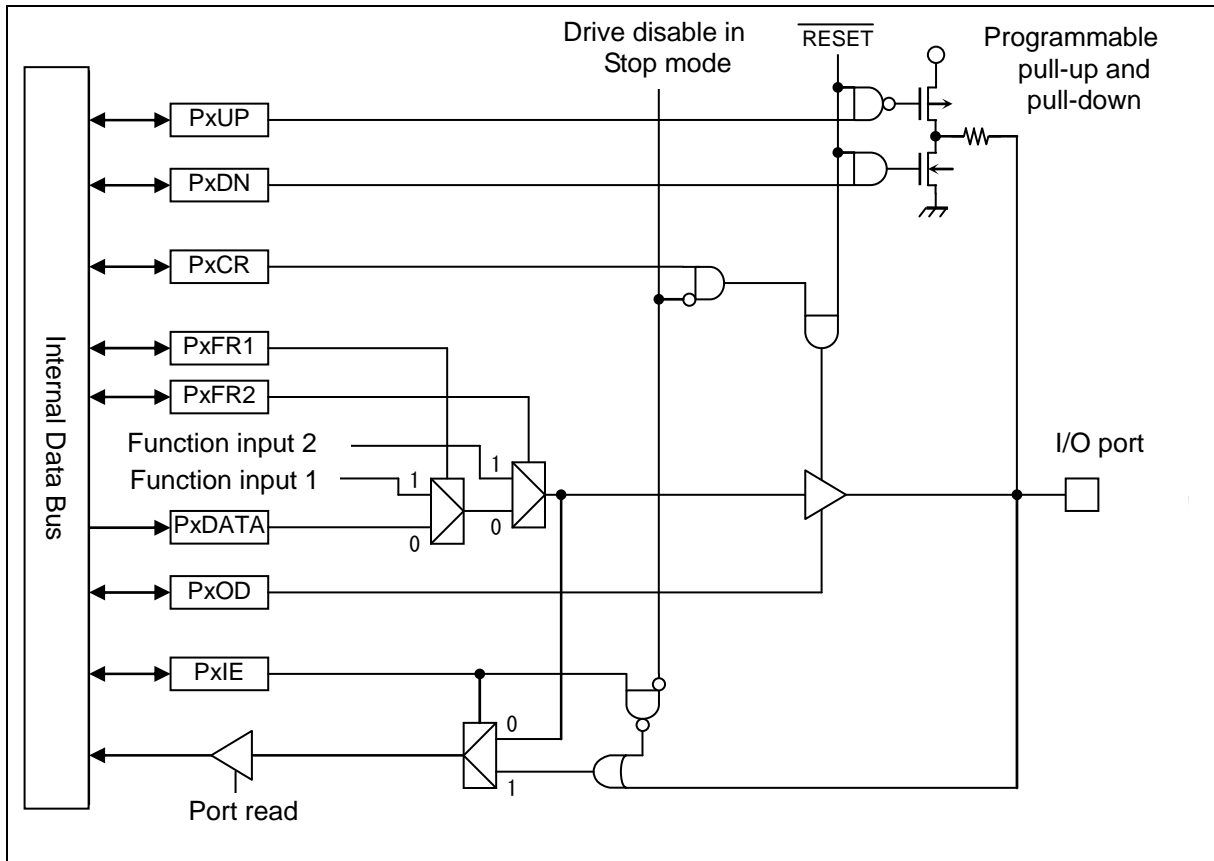
Fig 9-13 Type T13

Input/Output Ports

9.3.15 Type T14

Type T14 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data and interrupt input as well.
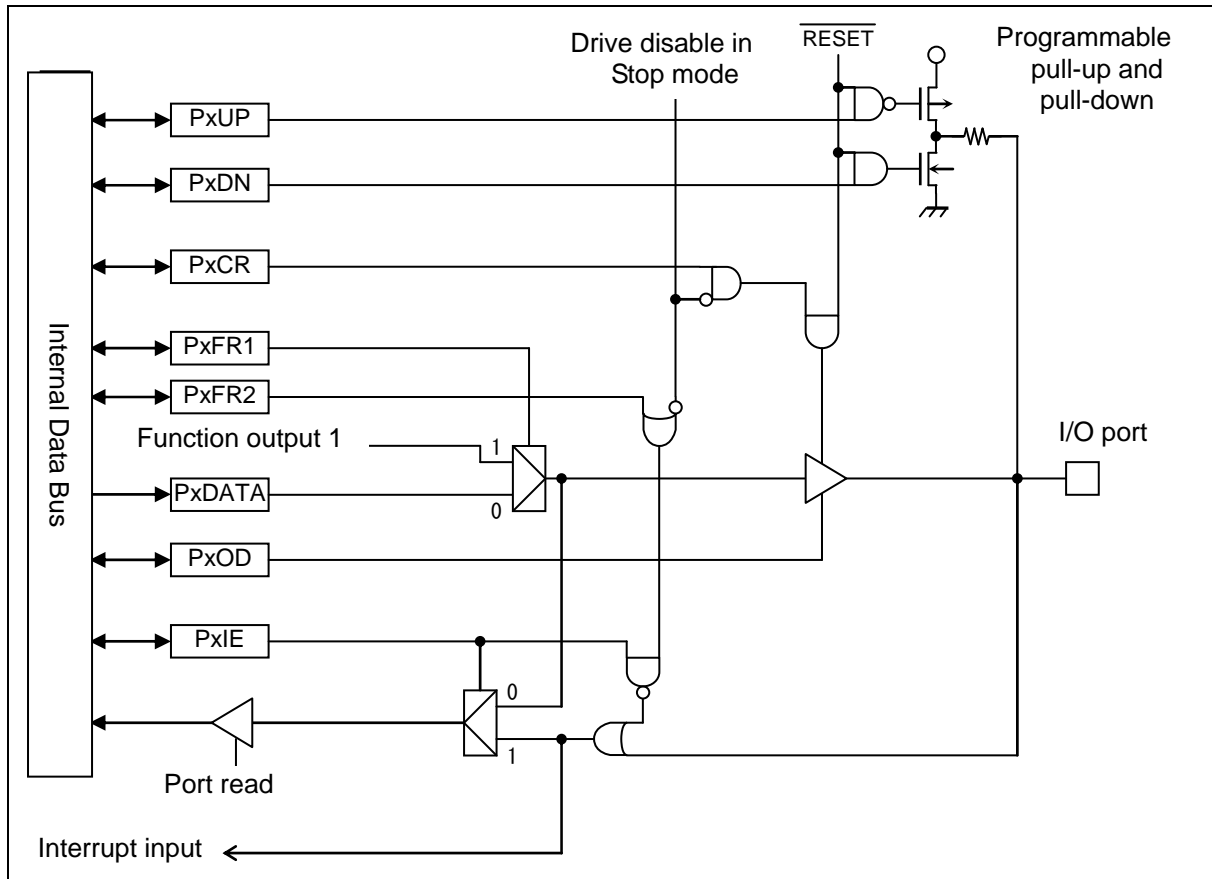
Fig 9-14 Type T14

9.3.16 Type T15

Type T15 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data and three input function data as well.
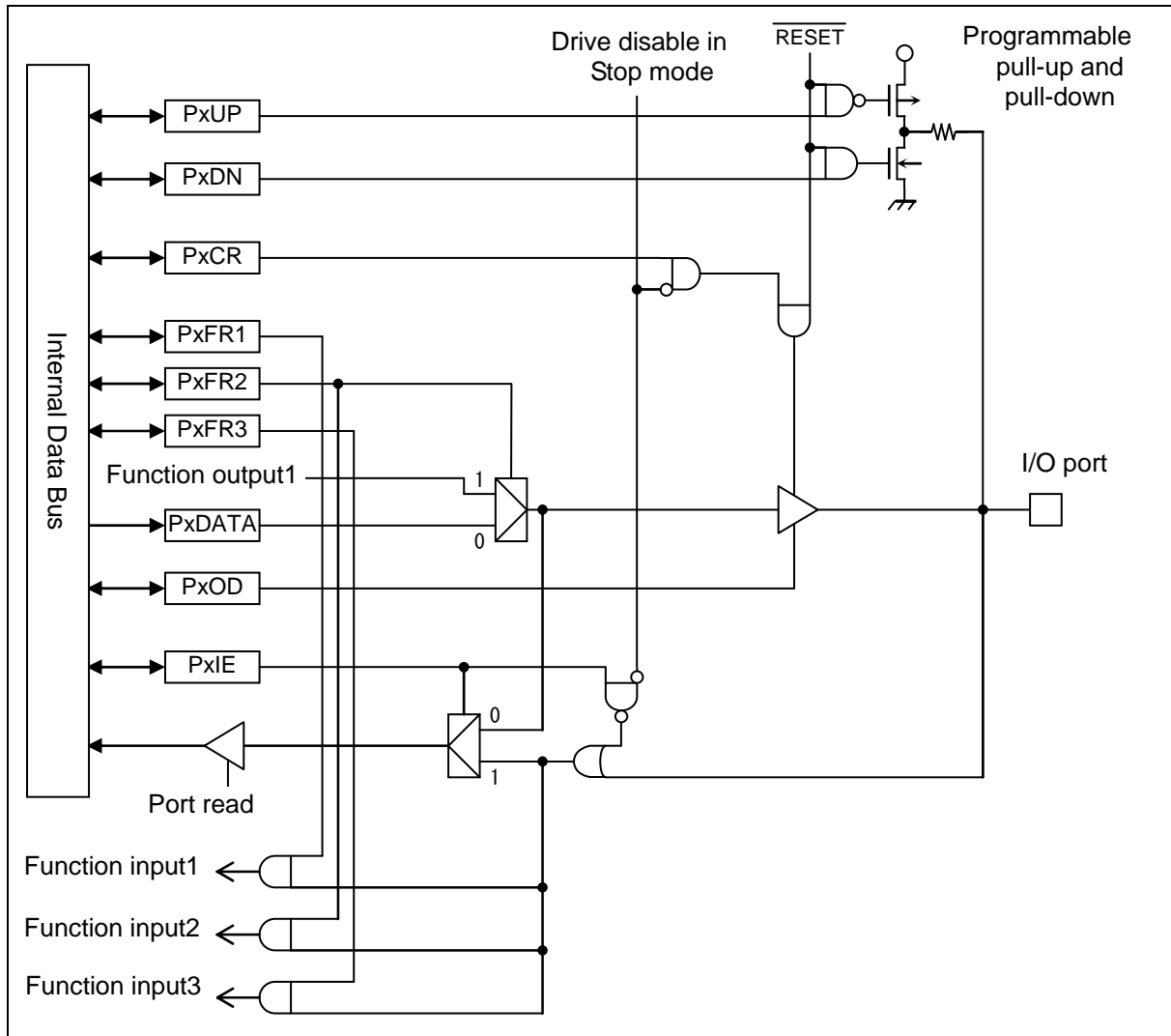
Fig 9-15 Type T15

Input/Output Ports

9.3.17 Type T16

Type T16 is a general-purpose input/output port with pull-up and pull-down. It is used to input analog signals for A/D converter as well.
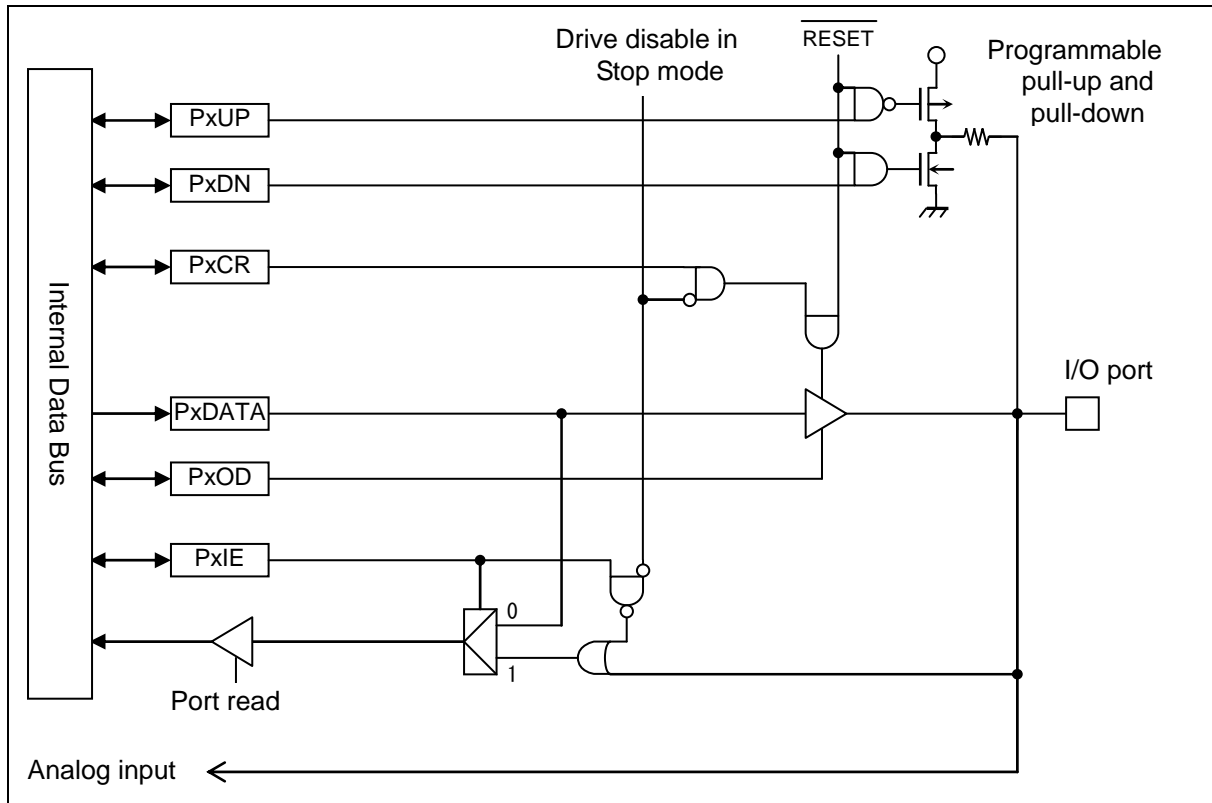
Fig 9-16 Type T16

9.3.18 Type T17

Type T17 is a general-purpose input/output port with pull-up and pull-down. It is used to input analog signals for A/D converter and interrupt input as well.
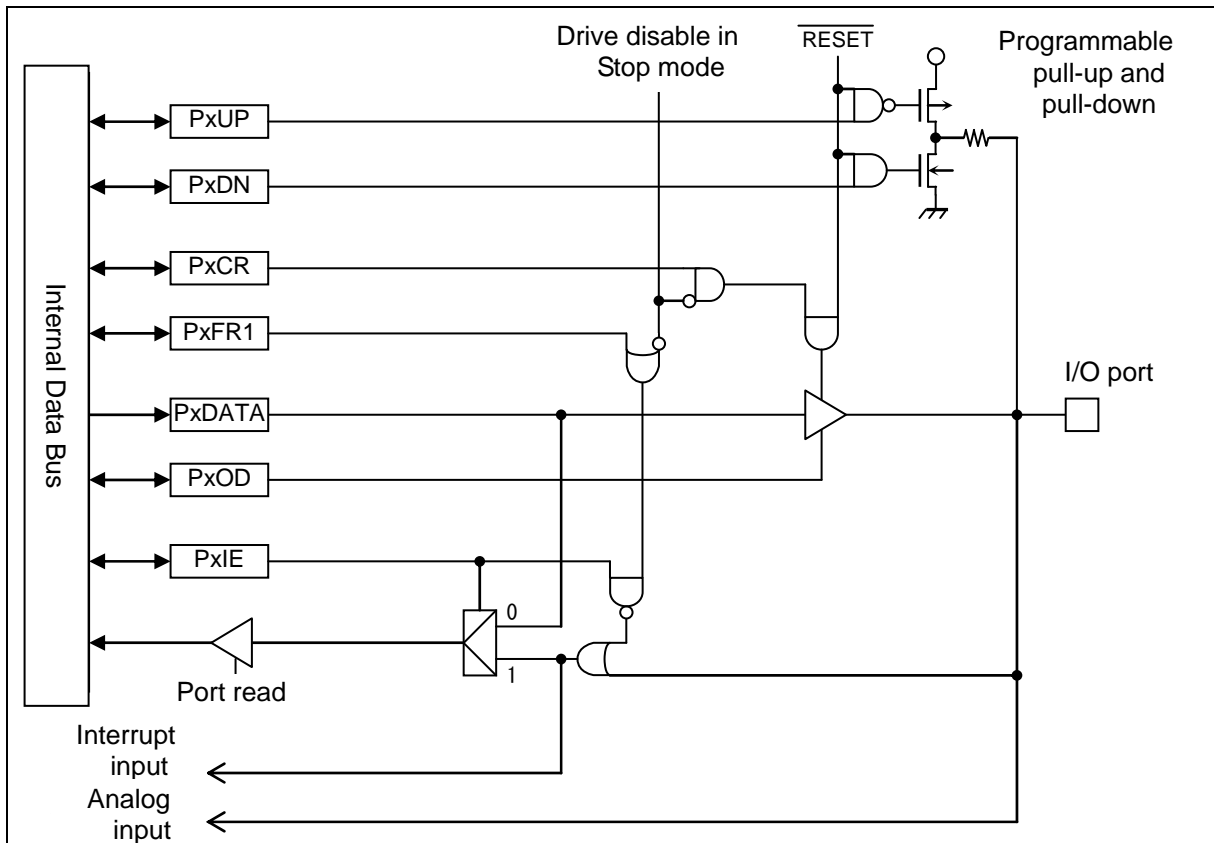
Fig 9-17 Type T17

9.3.19 Type T18

Type T18 is a general-purpose input/output port with pull-up and pull-down. It is used to output function data as well.
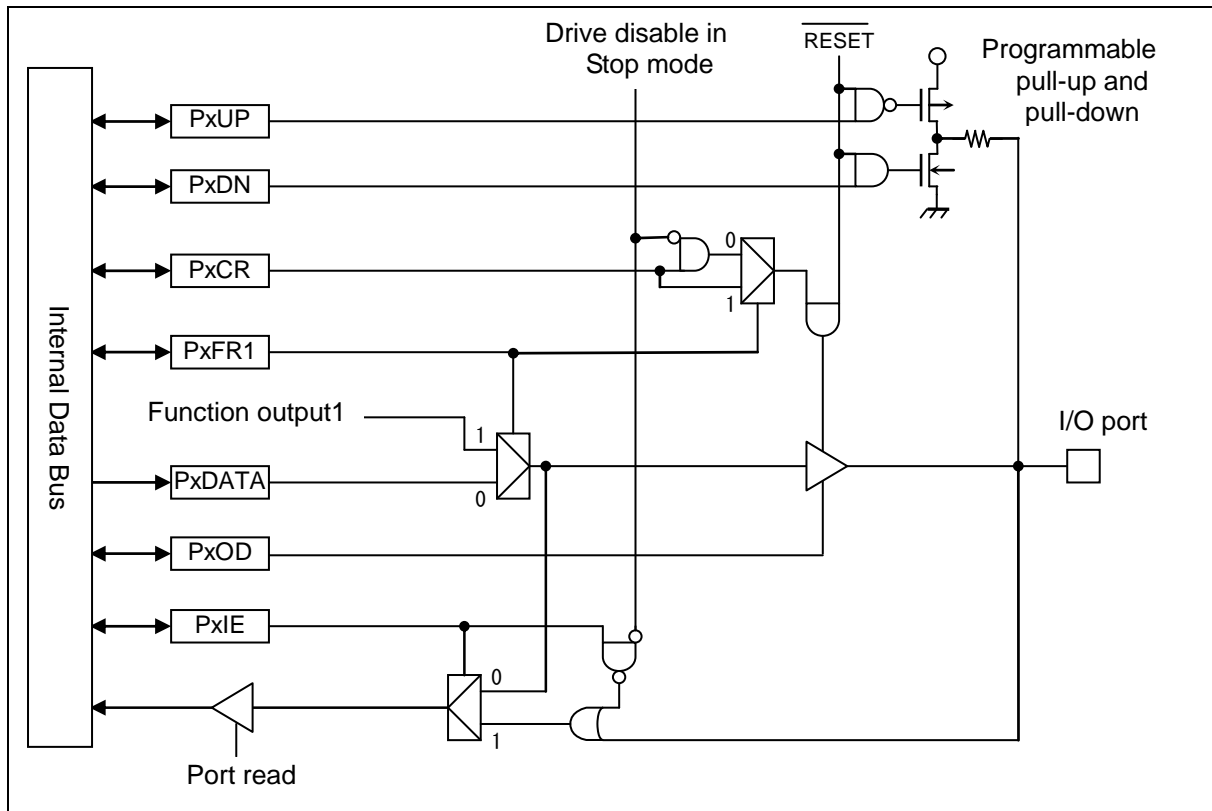
Fig 9-18 Type T18

9.3.20 Type T19

Type T19 is a general-purpose port with pull-up and pull-down. It is used to output function data as well.

The function output is controlled by an enable signal. If enabled, the function data is output.
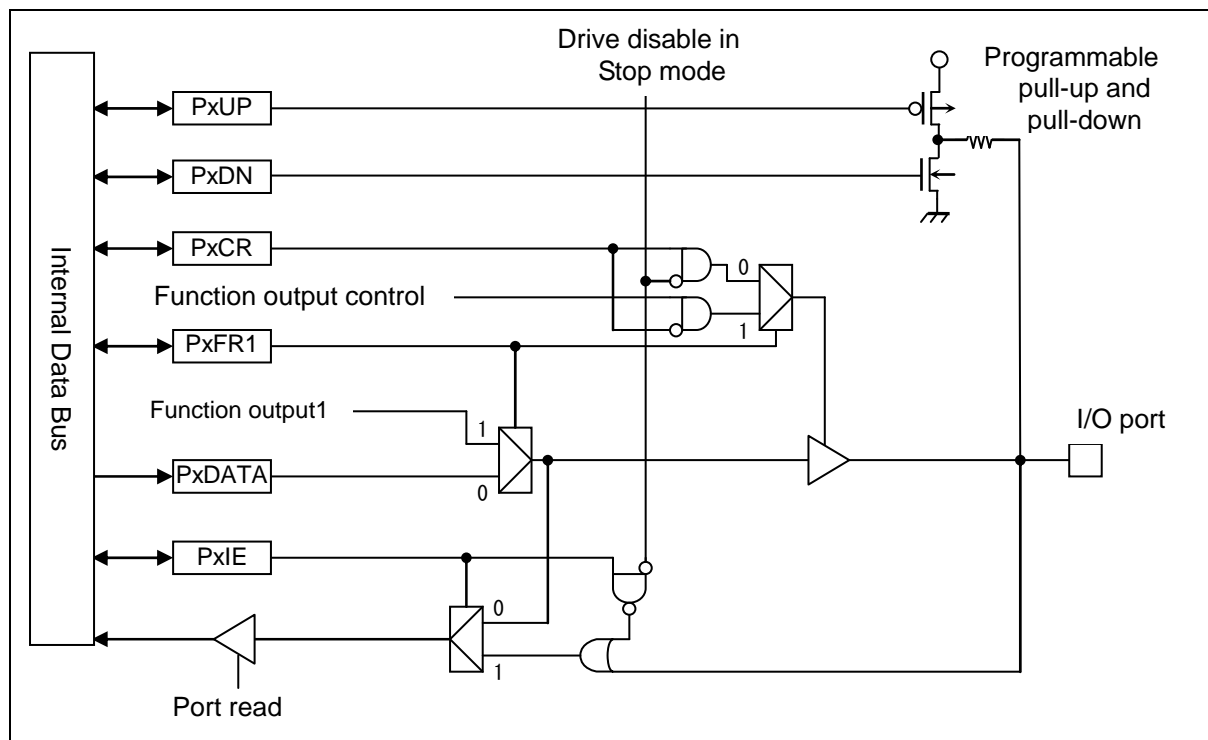
Fig 9-19 Type T19

9.3.21 Type T20

Type T20 is a general-purpose input/output port with pull-up and pull-down. It is used to input function data as well.

During reset, it functions as an input port for a BOOT signal.
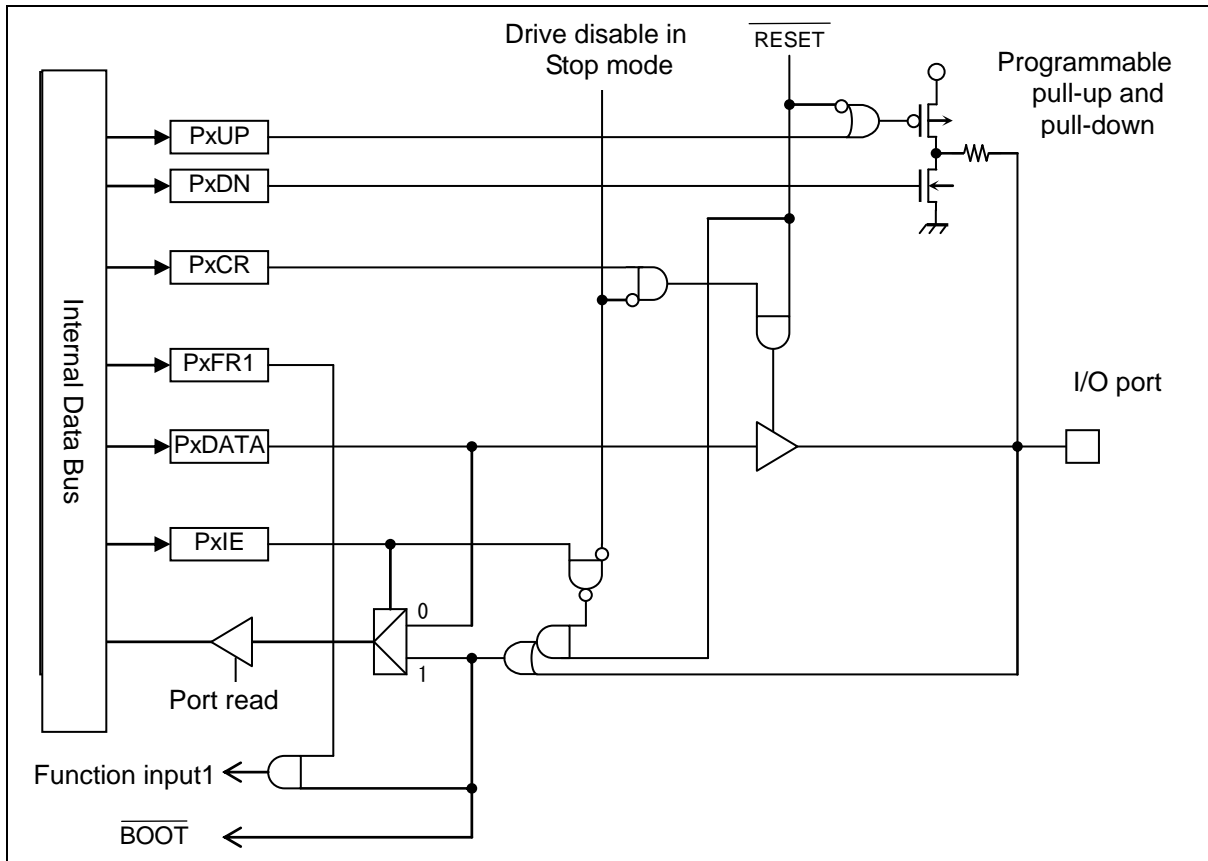
Fig 9-20 Type T20

9.3.22 Type T21

Type T21 is a general-purpose input/output port with pull-up and pull-down. It is used to high-speed oscillator1 (External) as well.
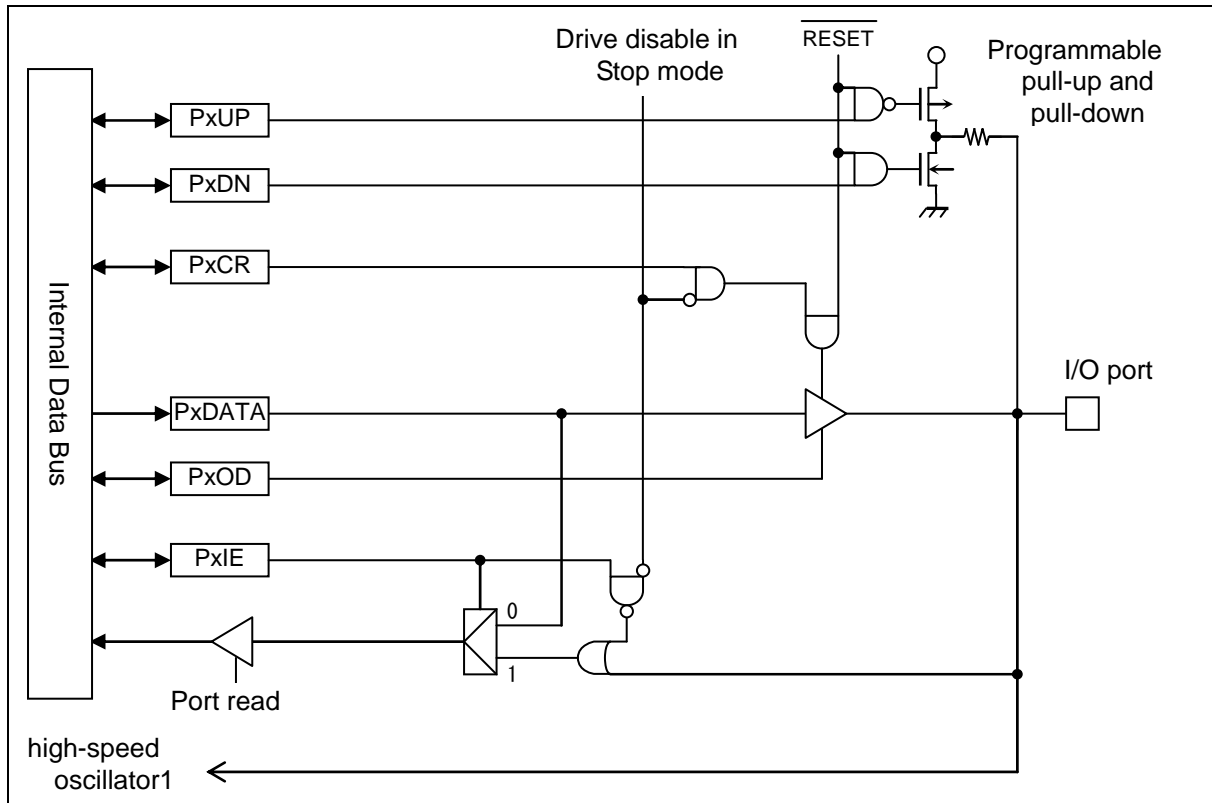
Fig 9.21 Type T21

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

10.1 Outline

TMPM372/3/4 has the eight channels multi-functional 16-bit timer/event counter. (TMRB0 through TMRB7) TMRBs operate in the following four operation modes:

- 16-bit interval timer mode

- 16-bit event counter mode

- 16-bit programmable square-wave output mode (PPG)

- External trigger programmable square-wave output mode (PPG)

(Note) Available operating mode depends on the product. Do not specify the non-available

operating mode in Table 10-1.

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

Pulse width measurement

One-shot pulse generation from an external trigger pulse

Table 10-1 Availability of operating mode in each product (○: Available, ✕: non-available)

Timer/counter	Operating mode	TMPM372	TMPM373	TMPM374
TMRB0	16-bit interval timer	○	○	○
	16-bit event counter	○	✕ (Note 2)	✕ (Note 3)
	16-bit PPG	○	✕ (Note 2)	✕ (Note 3)
TMRB1	16-bit interval timer	○	○	○
	16-bit event counter	○	○	✕ (Note 3)
	16-bit PPG	○	✕ (Note 2)	✕ (Note 3)
TMRB2	16-bit interval timer	○	○	○
	16-bit event counter	○	○	○
	16-bit PPG	○	✕ (Note 2)	✕ (Note 3)
TMRB3	16-bit interval timer	○	○	○
	16-bit event counter	○	○	○
	16-bit PPG	○	○	○
TMRB4	16-bit interval timer	○	○	○
	16-bit event counter	○	✕ (Note 2)	✕ (Note 3)
	16-bit PPG	○	✕ (Note 2)	✕ (Note 3)
TMRB5	16-bit interval timer	○	○	○
	16-bit event counter	✕ (Note 1)	✕ (Note 2)	✕ (Note 3)
	16-bit PPG	✕ (Note 1)	✕ (Note 2)	✕ (Note 3)
TMRB6	16-bit interval timer	○	○	○
	16-bit event counter	○	○	○
	16-bit PPG	○	○	○
TMRB7	16-bit interval timer	○	○	○
	16-bit event counter	○	○	○
	16-bit PPG	○	○	✕ (Note 3)

Note1: TMPM372 doesn't have TB5IN and TB5OUT pins.
Note2: TMPM373 doesn't have TB0IN, TB0OUT, TB1OUT, TB2OUT, TB4IN, TB4OUT, TB5IN and TB5OUT pins.
Note3: TMPM374 doesn't have TB0IN, TB0OUT, TB1IN, TB1OUT, TB2OUT, TB4IN, TB4OUT, TB5IN, TB5OUT
and TB7OUT pins.

10.2 Specification differences among channels

Channels (TMRB0 through TMRB7) operate independently and the functions are same except the differences as shown in Table 10-2 and Table 10-3. Therefore, the operational descriptions here are explained only for TMRB0.

Table 10-2 Differences in the Specifications of TMRB Modules (1) (○: Available, ✕: non-available)

Specification / Channel	External clock/capture trigger input pin or Timer flip-flop output pin	TMPM372	TMPM373	TMPM374
TMRB0	TB0IN (shared with PA0)	○	✕	✕
	TB0OUT (shared with PA1)	○	✕	✕
TMRB1	TB1IN (shared with PA2)	○	○	✕
	TB1OUT (shared with PA3)	○	✕	✕
TMRB2	TB2IN (shared with PE4)	○	○	○
	TB2OUT (shared with PE5)	○	✕	✕
TMRB3	TB3IN (shared with PE6)	○	○	○
	TB3OUT (shared with PE7)	○	○	○
TMRB4	TB4IN (shared with PA7)	○	✕	✕
	TB4OUT (shared with PE3)	○	✕	✕
TMRB5	TMPM372/373/374 don't have TB5IN and TB5OUT pins.	-	-	-
TMRB6	TB6IN (shared with PA6)	○	○	○
	TB6OUT (shared with PA5)	○	○	○
TMRB7	TB7IN (shared with PF0)	○	○	○
	TB7OUT (shared with PF1)	○	○	✕

Table 10-3 Differences in the Specifications of TMRB Modules (2)

Specification / Channel	Interrupt	
	Capture interrupt	TMRB interrupt
TMRB0	INTCAP00 INTCAP01	INTTB00 INTTB01
TMRB1	INTCAP10 INTCAP11	INTTB10 INTTB11
TMRB2	INTCAP20 INTCAP21	INTTB20 INTTB21
TMRB3	INTCAP30 INTCAP31	INTTB30 INTTB31
TMRB4	INTCAP40 INTCAP41	INTTB40 INTTB41
TMRB5	INTCAP50 INTCAP51	INTTB50 INTTB51
TMRB6	INTCAP60 INTCAP61	INTTB60 INTTB61
TMRB7	INTCAP70 INTCAP71	INTTB70 INTTB71

Note1: INTCAP01 and INTCAP41 are not available in TMPM373.
Note2: INTCAP01, INTCAP11 and INTCAP41 are not available in TMPM374.

10.3 Configuration

Each channel consists of a 16-bit up-counter, two 16-bit timer registers (one of which is double-buffered), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit.

Timer operation modes and the timer flip-flop are controlled by registers.

Note1: TMPM372 doesn't have TB5IN and TB5OUT pins.
Note2: TMPM373 doesn't have TB0IN, TB0OUT, TB1OUT, TB2OUT, TB4IN, TB4OUT, TB5IN and TB5OUT pins.
Note3: TMPM374 doesn't have TB0IN, TB0OUT, TB1IN, TB1OUT, TB2OUT, TB4IN, TB4OUT, TB5IN, TB5OUT and TB7OUT pins.
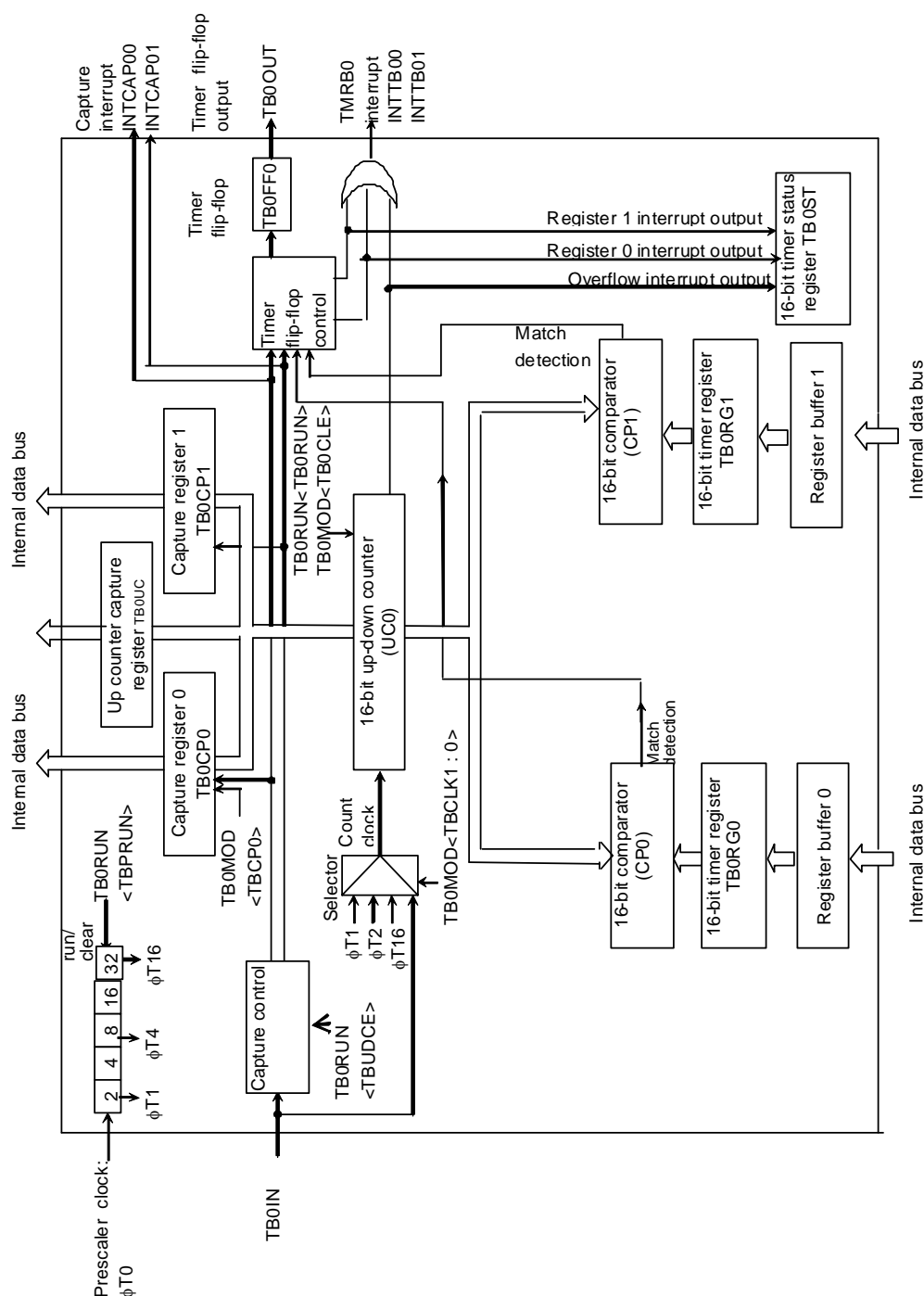
Fig. 10-1 TMRB0 Block Diagram (the same applies to channels 1 through7)

10.4 Registers

10.4.1 TMRB registers

Table 10-4 shows the register names and addresses of each channel.

Table 10-4 TMRB registers

Channel / Specification		TMRB0		TMRB1		TMRB2		TMRB3	
Register names (addresses)	Timer enable register	TB0EN	0x4001_0000	TB1EN	0x4001_0040	TB2EN	0x4001_0080	TB3EN	0x4001_00C0
	Timer RUN register	TB0RUN	0x4001_0004	TB1RUN	0x4001_0044	TB2RUN	0x4001_0084	TB3RUN	0x4001_00C4
	Timer control register	TB0CR	0x4001_0008	TB1CR	0x4001_0048	TB2CR	0x4001_0088	TB3CR	0x4001_00C8
	Timer mode register	TB0MOD	0x4001_000C	TB1MOD	0x4001_004C	TB2MOD	0x4001_008C	TB3MOD	0x4001_00CC
	Timer flip-flop control register	TB0FFCR	0x4001_0010	TB1FFCR	0x4001_0050	TB2FFCR	0x4001_0090	TB3FFCR	0x4001_00D0
	Timer status register	TB0ST	0x4001_0014	TB1ST	0x4001_0054	TB2ST	0x4001_0094	TB3ST	0x4001_00D4
	Interrupt mask register	TB0IM	0x4001_0018	TB1IM	0x4001_0058	TB2IM	0x4001_0098	TB3IM	0x4001_00D8
	Timer up counter register	TB0UC	0x4001_001C	TB1UC	0x4001_005C	TB2UC	0x4001_009C	TB3UC	0x4001_00DC
	Timer register	TB0RG0	0x4001_0020	TB1RG0	0x4001_0060	TB2RG0	0x4001_00A0	TB3RG0	0x4001_00E0
		TB0RG1	0x4001_0024	TB1RG1	0x4001_0064	TB2RG1	0x4001_00A4	TB3RG1	0x4001_00E4
	Capture register	TB0CP0	0x4001_0028	TB1CP0	0x4001_0068	TB2CP0	0x4001_00A8	TB3CP0	0x4001_00E8
		TB0CP1	0x4001_002C	TB1CP1	0x4001_006C	TB2CP1	0x4001_00AC	TB3CP1	0x4001_00EC

Channel / Specification		TMRB4		TMRB5		TMRB6		TMRB7	
Register names (addresses)	Timer enable register	TB4EN	0x4001_0100	TB5EN	0x4001_0140	TB6EN	0x4001_0180	TB7EN	0x4001_01C0
	Timer RUN register	TB4RUN	0x4001_0104	TB5RUN	0x4001_0144	TB6RUN	0x4001_0184	TB7RUN	0x4001_01C4
	Timer control register	TB4CR	0x4001_0108	TB5CR	0x4001_0148	TB6CR	0x4001_0188	TB7CR	0x4001_01C8
	Timer mode register	TB4MOD	0x4001_010C	TB5MOD	0x4001_014C	TB6MOD	0x4001_018C	TB7MOD	0x4001_01CC
	Timer flip-flop control register	TB4FFCR	0x4001_0110	-	-	TB6FFCR	0x4001_0190	TB7FFCR	0x4001_01D0
	Timer status register	TB4ST	0x4001_0114	TB5ST	0x4001_0154	TB6ST	0x4001_0194	TB7ST	0x4001_01D4
	Interrupt mask register	TB4IM	0x4001_0118	TB5IM	0x4001_0158	TB6IM	0x4001_0198	TB7IM	0x4001_01D8
	Timer up counter register	TB4UC	0x4001_011C	TB5UC	0x4001_015C	TB6UC	0x4001_019C	TB7UC	0x4001_01DC
	Timer register	TB4RG0	0x4001_0120	TB5RG0	0x4001_0160	TB6RG0	0x4001_01A0	TB7RG0	0x4001_01E0
		TB4RG1	0x4001_0124	TB5RG1	0x4001_0164	TB6RG1	0x4001_01A4	TB7RG1	0x4001_01E4
	Capture register	TB4CP0	0x4001_0128	TB5CP0	0x4001_0168	TB6CP0	0x4001_01A8	TB7CP0	0x4001_01E8
		TB4CP1	0x4001_012C			TB6CP1	0x4001_01AC	TB7CP1	0x4001_01EC

Note1: Do not access to TB0FFCR, TB1FFCR, TB2FFCR and TB4FFCR in TMPM373.
Note2: Do not access to TB0FFCR, TB1FFCR, TB2FFCR, TB4FFCR and TB7FFCR in TMPM373.
Note3: The content s of TB0CP1 and TB4CP1 in TMPM373 are invalid.
Note4: The content s of TB0CP1, TB1CP1 and TB4CP1 in TMPM374 are invalid.

10.4.1.1   TMRBn enable register (channels 0 through 7)

TMRBn enable register (n=0 to 7)

TBnEN
(0x4001_0xx0)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol	TBnEN	TBHALT						
Read/Write	R/W	R/W				R		
After reset	0	0				0		
Function	TMRBn operation 0: Disabled 1: Enabled	Clock operation during debug HALT 0: run 1: stop				"0" is read.		

<TBHALT>: Specifies the TMRB clock setting to run or stop when the debug tool transits to HALT mode while in use.

<TBnEN>: Specifies the TMRBn operation. When the operation is disabled, no clock is supplied to the other registers in the TMRBn module. This can reduce power consumption. (This disables reading from and writing to the other registers.)

> To use the TMRBn, enable the TMRBn operation (set to "1") before programming each register in the TMRBn module.

After the TMRBn operation is executed and then disabled, the settings will be maintained in each registers.

10.4.1.2 TMRB RUN register (channels 0 through 7)

TMRBn RUN register (n=0 to 7)

TBnRUN
(0x4001_0xx4)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol						TBnPRUN		TBnRUN
Read/Write	R					R/W	R	R/W
After reset	0					0	0	0
Function	"0" is read.					Timer Run/Stop Control 0: Stop & clear 1: Count * The bit 1 can be read as "0."		

＜TBnRUN＞ :Controls the TMRBn count operation.

    0: Stop counting and the counter is cleared to "0".

    1: Start counter

(Note) To start couting by the external trigger, the TBnRUN bit must be set to 1.

＜TBnPRUN＞:Controls the TMRBn prescaler operation.

    0: Stop prescaler operation and the prescaler is cleared to "0".

    1: Start prescaler operation.

10.4.1.3   TMRB control register (channels 0 through 7)

TMRBn control register (n=0 to 7)

TBnCR (0x4001_0xx8)		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		7	6	5	4	3	2	1	0
	bit Symbol	TBnWBF				I2TBn		TRGSELn	CSSELn
	Read/Write	R/W	R/W	R/W	R	R/W	R	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	Double Buffer 0: Disabled 1: Enabled	Write "0".	Write "0".	"0" is read.	IDLE 0:Stop 1:Operation	"0" is read.	External Trigger select 0: Rising edge 1: Falling edge	Counter Start select 0: Software start 1: External trigger

<CSSELn>: Selects how the timer starts counting.

      0: Select software for timer count start.

      1: Select external trigger for timer count start.

> Note 1: Do not set <CSSELn> to "1" for TMRB5 in TMPM372.
> Note 2: Do not set <CSSELn> to "1" for TMRB0, TMRB4 and TMRB5 in TMPM373.
> Note 3: Do not set <CSSELn> to "1" for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374.

<TRGSELn>: Selects the active edge of the external trigger signal.

      0: Select rising edge of external trigger.

      1: Select falling edge of external trigger.

> Note 1: Do not set <TRGSELn> to "1" for TMRB5 in TMPM372.
> Note 2: Do not set <TRGSELn> to "1" for TMRB0, TMRB4 and TMRB5 in TMPM373.
> Note 3: Do not set <TRGSELn> to "1" for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374.

＜I2TBn＞:Controls the clock keep/ stop operation during the IDLE mode.

      0: Stop the clock.

      1: Keep clock operation during IDLE mode.

＜TBWBF＞:Controls the enabling/disabling of double buffering.

      0: Disable Double Buffer.

      1: Enable Double Buffer.

> (Note) TBnCR resister must not be changed during Timer operation TBnRUN<TERUN>=1".

10.4.1.4 TMRB mode register (channels 0 thorough 7)

TMRBn mode register(n=0 to 7)

TBnMOD
(0x4001_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol		TBnRSWR	TBnCP0	TBnCPM1	TBnCPM0	TBnCLE	TBnCLK1	TBnCLK0
Read/Write	R	R/W	W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Writes to timer registers 0 and 1 (when double buffering is enabled) 0: Can be written separately 1: Must be written simultaneously	Software capture control 0: Software capture 1: Don't care	Capture timing 00: Disable 01: TBnIN ↑ 10: TBnIN ↑ TBnIN ↓ 11: Disable		Up-counter clear control 0: Clear/disable 1: clear/enable	Source clock select 00: TBnIN pin input 01: φT1 10: φT4 11: φT16	

＜TBnCLK1:0＞:Selects the TMRBn timer count clock.
　　　　　00: select TBnIN input pin
　　　　　01: select φT1　　(1/2φT0)
　　　　　10: select φT4　　(1/8φT0)
　　　　　11: select φT16　(1/32φT0)

> Note 1: Do not set "00" to <TBnCLK1:0> for TMRB5 in TMPM372.
> Note 2: Do not set "00" to <TBnCLK1:0> for TMRB0, TMRB4 and TMRB5 in TMPM373.
> Note 3: Do not set "00" to <TBnCLK1:0> for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374.

＜TBnCLE＞:Clears and controls the TMRBn up-counter.
　　　　　"0": Disables clearing of the up-counter.
　　　　　"1": Clears up-counter if there is a match with timer register 1 (TBnRG1).
＜TBnCPM1:0＞:Specifies TMRBn capture timing.
　　　　　"00": Capture disable
　　　　　"01": Takes count values into capture register 0 (TBnCP0) upon rising of TBnIN pin input.
　　　　　"10": Takes count values into capture register 0 (TBnCP0) upon rising of TBnIN pin input.
　　　　　　　　Takes count values into capture register 1 (TBnCP1) upon falling of TBnIN pin input.
　　　　　"11": .Capture disable

> Note 1: Always write "00" to <TBnCPM1:0> for TMRB5 in TMPM372.
> Note 2: Always write "00" to <TBnCPM1:0> for TMRB0, TMRB4 and TMRB5 in TMPM373.
> Note 3: Always write "00" to <TBnCPM1:0> for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374.

＜TBnCP0＞:Captures count values by software and takes them into capture register 0 (TBnCP0).
<TBnRSWR>:   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

(Note 1) The value read from bit 5 of TBnMOD is "1".

(Note 2) TBnMOD register must not be changed during Timer operation TBnRUN<TBRUN>="1".

10.4.1.5   TMRB flip-flop control register (channels 0 through 7)

TMRBn flip-flop control register (n=0 to 7)

TBnFFCR (0x4001_0xx0)		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		7	6	5	4	3	2	1	0
	bit Symbol			TBnC1T1	TBnC0T1	TBnE1T1	TBnE0T1	TBnFF0C1	TBnFF0C0
	Read/Write	R		R/W				R/W	
	After reset	1	1	0	0	0	0	1	1
	Function	"11" is read.		TBnFF0 reverse trigger 0: Disable trigger 1: Enable trigger				TBnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * This is always read as "11."	
				When the up-counter value is taken into TBnCP1	When the up-counter value is taken into TBnCP0	When the up-counter matches TBnRG1	When the up-counter matches TBnRG0		

＜TBnFF0C1:0＞:Controls the timer flip-flop.

    00 : Reverses the value of TBnFF0

    01 : Sets TBnFF0 to "1".

    10 : Clears TBnFF0 to "0".

    11 :Don't care

＜TBnE0T1＞:Reverses the timer flip-flop when the up-counter matches the timer register 0 (TBnRG0).

    0: TBnFF0   not reverse

    1: TBnFF0   reverse

＜TBnE1T1＞:Reverses the timer flip-flop when the up-counter matches the timer register 1 (TBnRG1).

    0: TBnFF0   not reverse

    1: TBnFF0   reverse

＜TBnC0T1＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 0 (TBnCP0).

    0: TBnFF0   not reverse

    1: TBnFF0   reverse

＜TBnC1T1＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 1 (TBnCP1).

    0: TBnFF0   not reverse

    1: TBnFF0   reverse

Note 1: TBnFFCR register must not be changed during Timer operation TBnRUN<TBRUN>="1".

Note 2: Do not access to TBnFFCR for TMRB5 in TMPM372.

Note 3: Do not access to TBnFFCR for TMRB0, TMRB1, TMRB2, TMRB4 and TMRB5 in TMPM373.

Note 4: Do not access to TBnFFCR for TMRB0, TMRB1, TMRB2, TMRB4, TMRB5 and TMRB7 in TMPM374.

10.4.1.6 TMRB status register (channels 0 through 7)

TMRBn status register (n=0 to 7)

TBnST
(0x4001_0xx4)

		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		7	6	5	4	3	2	1	0
	bit Symbol						INTTBOFn	INTTBn1	INTTBn0
	Read/Write			R				R	
	After reset			0			0	0	0
	Function			"0" is read.			0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

＜INTTBn0＞:Interrupt generated status for a match with timer register 0 (TBnRG0)
       0: No interrupt generated
       1: Interrupt generated
＜INTTBn1＞:Interrupt generated status for a match with timer register 1 (TBnRG1)
       0: No interrupt generated
       1: Interrupt generated
＜INTTBOFn＞:Interrupt generated status for an up-counter overflow occurs
       0: No interrupt generated
       1: Interrupt generated

(Note) If any interrupt is generated, the flag that corresponds to the interrupt is set to TBnST and the generation of interrupt is notified to the CPU.
The flag is cleared by reading the TBnST register.

10.4.1.7 TMRB interrupt mask register (channels 0 through 7)

TMRBn interrupt mask register (n=0 to 7)

TBnIM
(0x4001_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol						TBIMOFn	TBIMn1	TBIMn0
Read/Write	R					R/W		
After reset	0					0	0	0
Function	"0" is read					0: No Interrupt mask 1: Interrupt is masked	0: No Interrupt mask 1: Interrupt is masked	0: No Interrupt mask 1: Interrupt is masked

＜TBIMn0＞: Interrupt mask for a match with timer register 0 (TBnRG0)
        0: No interrupt mask
        1: Interrupt is masked
＜TBIMn1＞:Interrupt mask for a match with timer register 1 (TBnRG1).
        0: No interrupt mask
        1: Interrupt is masked
＜TBIMOFn＞:Interrupt mask for an up counter overflow.
        0: No interrupt mask
        1: Interrupt is masked

(Note) Even in case TBnIM set interrupt Mask, TBnST status register have an interrupt requested status.

10.4.1.8 TMRB read capture register (channels 0 through 7)

TBnUC0 read capture register (n=0 to 7)

TBnUC0 (0x4001_0xxC)		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
		15	14	13	12	11	10	9	8
	bit Symbol	UCn15	UCn14	UCn13	UCn12	UCn11	UCn10	UCn9	UCn8
	Read/Write	R							
	After reset	0							
	Function	Data obtained by read capture: 15-8 bit data							
		7	6	5	4	3	2	1	0
	bit Symbol	UCn7	UCn6	UCn5	UCn4	UCn3	UCn2	UCn1	UCn0
	Read/Write	R							
	After reset	0							
	Function	Data obtained by read capture: 7-0 bit							

<UCn15-0> : Captured Up-counter value.

10.4.1.9 TMRB timer register (channels 0 through 7)

TBnRG0 timer register (n=0 to 7)

TBnRG0
(0x4001_0xx0)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBnRG015	TBnRG014	TBnRG013	TBnRG012	TBnRG011	TBnRG010	TBnRG09	TBnRG08
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBnRG07	TBnRG06	TBnRG05	TBnRG04	TBnRG03	TBnRG02	TBnRG01	TBnRG00
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

<TBnRG015-000> : 16bit Compare value0 with Up- counter

TBnRG1 timer register (n=0~7)

TBnRG1
(0x4001_0xx4)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBnRG115	TBnRG114	TBnRG113	TBnRG112	TBnRG111	TBnRG110	TBnRG19	TBnRG18
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBnRG17	TBnRG16	TBnRG15	TBnRG14	TBnRG13	TBnRG12	TBnRG11	TBnRG10
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

<TBnRG115-100> : 16bit Compare value1 with Up- counter

10.4.1.10 TMRB capture register (channels 0 through 7)

TBnCP0capture register (n=0 to 7)

TBnCP0
(0x4001_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBnCP015	TBnCP014	TBnCP013	TBnCP012	TBnCP011	TBnCP010	TBnCP09	TBnCP08
Read/Write	R							
After reset	0							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBnCP07	TBnCP06	TBnCP05	TBnCP04	TBnCP03	TBnCP02	TBnCP01	TBnCP00
Read/Write	R							
After reset	0							
Function	Timer capture value: 7-0 bit data							

<TBnCP015-000>: 16bit up-counter capture value0

TBnCP1 capture register (n=0 to 7)

TBnCP1
(0x4001_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBnCP115	TBnCP114	TBnCP113	TBnCP112	TBnCP111	TBnCP110	TBnCP19	TBnCP18
Read/Write	R							
After reset	0							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBnCP17	TBnCP16	TBnCP15	TBnCP14	TBnCP13	TBnCP12	TBnCP11	TBnCP10
Read/Write	R							
After reset	0							
Function	Timer capture value: 7-0 bit data							

<TBnCP115-100>: 16bit up-counter capture value1

Note 1: TBnCP1 for TMRB5 in TMPM372 is invalid because there is no TB5IN pin.
Note 2: TBnCP1 for TMRB0, TMRB4 and TMRB5 in TMPM373 is invalid because the TMPM373 doesn't
have TB0IN, TB4IN and TB5IN pins.
Note 3: TBnCP1 for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374 is invalid because TMPM374
doesn't have TB0IN, TB1IN, TB4IN and TB5IN pin.

10.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 9-2 and Table 9-3. Therefore, the operational descriptions here are only for channel 0.

10.5.1 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter UC. The prescaler input clock ϕT0 is fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by SYSCR1<PRCLK2:0> in the CG. The peripheral clock, fperiph, is either fgear, a clock selected by SYSCR1<FPSEL> in the CG, or fc, which is a clock before it is divided by the clock gear.

The operation or the stop of a prescaler is set with TB0RUN<TB0PRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 9-5 shows prescaler output clock resolutions.

Table 10-5 Prescaler Output Clock Resolutions @fc = 80MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution		
			ϕT1	ϕT4	ϕT16
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025µs)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)
		001 (fperiph/2)	$fc/2^2$ (0.05µs)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)
		010 (fperiph/4)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		011 (fperiph/8)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		100 (fperiph/16)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		101 (fperiph/32)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.05µs)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)
		001 (fperiph/2)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		010 (fperiph/4)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		011 (fperiph/8)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		100 (fperiph/16)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
		101 (fperiph/32)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)	$fc/2^{11}$ (25.6µs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		001 (fperiph/2)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		010 (fperiph/4)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		011 (fperiph/8)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
		100 (fperiph/16)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)	$fc/2^{11}$ (25.6µs)
		101 (fperiph/32)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)	$fc/2^{12}$ (51.2µs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		001 (fperiph/2)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		010 (fperiph/4)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
		011 (fperiph/8)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)	$fc/2^{11}$ (25.6µs)
		100 (fperiph/16)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)	$fc/2^{12}$ (51.2µs)
		101 (fperiph/32)	$fc/2^9$ (6.4µs)	$fc/2^{11}$ (25.6µs)	$fc/2^{13}$ (102.4µs)
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025µs)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)
		001 (fperiph/2)	$fc/2^2$ (0.05µs)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)
		010 (fperiph/4)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		011 (fperiph/8)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		100 (fperiph/16)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		101 (fperiph/32)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
	100 (fc/2)	000 (fperiph/1)	—	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)
		001 (fperiph/2)	$fc/2^2$ (0.05µs)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)
		010 (fperiph/4)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		011 (fperiph/8)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		100 (fperiph/16)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		101 (fperiph/32)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
	101 (fc/4)	000 (fperiph/1)	—	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)
		001 (fperiph/2)	—	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)
		010 (fperiph/4)	$fc/2^3$ (0.1µs)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		011 (fperiph/8)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		100 (fperiph/16)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		101 (fperiph/32)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)
	110 (fc/8)	000 (fperiph/1)	—	—	$fc/2^5$ (0.4µs)
		001 (fperiph/2)	—	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)
		010 (fperiph/4)	—	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)
		011 (fperiph/8)	$fc/2^4$ (0.2µs)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)
		100 (fperiph/16)	$fc/2^5$ (0.4µs)	$fc/2^7$ (1.6µs)	$fc/2^9$ (6.4µs)
		101 (fperiph/32)	$fc/2^6$ (0.8µs)	$fc/2^8$ (3.2µs)	$fc/2^{10}$ (12.8µs)

(Note 1)	The prescaler output clock ϕTn must be selected as $\phi Tn < fsys$. (ϕTn is slower than fsys).
(Note 2)	Do not change the clock gear while the timer is operating.
(Note 3)	"—" denotes a setting prohibited.

10.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

   UC source clock, specified by TB0MOD<TB0CLK1:0>, can be selected from either three types ϕT1, ϕT4 and ϕT16 of prescaler output clock or the external clock of the TB0IN pin.

- Count start/ stop

   Counter operation is specified by TB0RUN<TB0RUN>. UC starts counting if <TB0RUN> = "1", and stops counting and clears counter value if <TB0RUN> = "0".

- Timing to clear UC

   1) When a compare match is detected

   By setting TB0MOD<TB0CLE> = "1", UC is cleared in case the comparator detects a match between counter value and the value set in TB0RG1. UC operates as a free-running counter if TB0MOD<TB0CLE> = "0".

   2) When UC stops

   UC stops counting and clears counter value if TB0RUN <TB0RUN> = "0".

- UC overflow

   If UC overflow occurs, the INTTB00 overflow interrupt is generated.

10.5.3 Timer registers (TB0RG0, TB0RG1)

TB0RG0 and TB0RG1 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.

TB0RG0 andTB0RG1 consist of the double-buffered configuration which is paired with register buffers. The double buffering is disabled in the initial state.

Controlling double buffering disable or enable is specified by TB0CR<TBWBF> bit. If the <TBWBF> is set to "0", the double buffering becomes disabled. If the <TBWBF> is set to "0", it becomes enabled. When the double buffering is enabled, a data transfer from the register buffer to the timer register (Tb0RG0/1) is done in the case that UC is matched with TB0RG1. When the counter is stopped even if the double buffering is enabled, the double buffering operates as a single buffer, and an immediate data can be written to the TB0RG0 and TB0RG1.

TB0RG0/ TB0RG1 and the register buffers are assigned to the same address. If the <TB0WBF> is set to "0," the same value is written to TB0RG0, TB0RG1 and each register buffer; if the <TB0WBF> is set to "1," the value is only written to each register buffer. Therefore, in order to write an initial value to the timer register, the register buffers must be set to "disable". Then set <TB0WBF> = "1"and write the following data to the register.

- Interrupt
  INTTB00 is generated by UC count value matching with TB0RG0 value.
  INTTB01 is generated by UC count value matching with TB0RG1 value.

10.5.4 Capture Control

This is a circuit that controls the timing to latch UC up-counter values into the TB0CP0 and TB0CP1 capture registers. The timing to latch data is specified by TB0MOD <TB0CPM1:0>.

Software can also be used to capture values from the UC up-counter into the capture register; specifically, UC values are taken into the TB0CP0 capture register each time "0" is written to TB0MOD<TB0CP0>. To use this capability, the prescaler must be running (TB0RUN<TB0PRUN> = "1").

10.5.5 Capture Registers (TB0CP0, TB0CP1)

These are 16-bit registers for latching values from the UC up-counter. To read data from the capture register, use a 16-bit data transfer instruction or read in the order of low-order bits followed by high-order bits.

> Note 1: TBnCP1 for TMRB5 in TMPM372 is invalid because there is no TB5IN pin.
> Note 2: TBnCP1 for TMRB0, TMRB4 and TMRB5 in TMPM373 is invalid because the TMPM373 doesn't have TB0IN, TB4IN and TB5IN pins.
> Note 3: TBnCP1 for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374 is invalid because TMPM374 doesn't have TB0IN, TB1IN, TB4IN and TB5IN pin.

10.5.6 Up-counter capture register (TB0UC)

Other than the capturing functions shown above, the current count value of the UC can be captured by reading the TB0UC registers.

10.5.7 Comparators (CP0,CP1)

These are 16-bit comparators for detecting a match by comparing set values of the UC up-counter with set values of the TB0RG0 and TB0RG1 timer registers. If a match is detected, INTTB00 and INTTB01 are generated.

10.5.8 Timer Flip-flop (TB0FF0)

The timer flip-flop (TB0FF0) 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 TB0FFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>.

The value of TB0FF0 becomes undefined after a reset. The flip-flop can be reversed by writing "00" to TB0FFCR<TB0FF0C1:0>. It can be set to "1" by writing "01," and can be cleared to "0" by writing "10."

The TB0FF0 value can be output to the timer output pin:TB0OUT (shared with PA1). To enable timer output, the port A related registers PACR and PAFR1 must be programmed beforehand.

> Note 1: Do not access to TBnFFCR for TMRB5 in TMPM372.
> Note 2: Do not access to TBnFFCR for TMRB0, TMRB1, TMRB2, TMRB4 and TMRB5 in TMPM373.
> Note 3: Do not access to TBnFFCR for TMRB0, TMRB1, TMRB2 TMRB4 TMRB5 and TMRB7 In TMPM374.

10.5.9 Capture interrupt (INTCAP00, INTCAP01)

Interrupts INTCAP00 and INTCAP01 can be generated at the timing of latching values from the UC up-counter into the TB0CP0 and TB0CP1 capture registers. The interrupt setting is specified by the CPU.

> Note 1: INTCAPn1 for TMRB5 in TMPM372 is invalid because there is no TB5IN pin.
> Note 2: INTCAPn1 for TMRB0, TMRB4 and TMRB5 in TMPM373 is invalid because the TMPM373 doesn't have TB0IN, TB4IN and TB5IN pins.
> Note 3: INTCAPn1 for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374 is invalid because TMPM374 doesn't have TB0IN, TB1IN, TB4IN and TB5IN pin.

10.6 Description of Operations for Each Mode

10.6.1 16-bit Interval Timer Mode

-Generating interrupts at periodic cycles

To generate the INTTB01 interrupt, specify a time interval in the TB0RG1 timer register.

Same as TB0RG0, INTTB01 interrupt is generated by setting different interval time value to TB0RG1 timer resister,

10.6.2 16-bit Event Counter Mode

It is possible to make it the event counter by using an input clock as an external clock (TB0IN pin input).

The up-counter counts up on the rising edge of TB0IN pin input. It is possible to read the count value by capturing value using software and reading the captured value.

To use it as an event counter, put the prescaler in a "RUN" state (TB0RUN<TB0PRUN> = "1").

Note 1: 16-bit Event Counter mode for TMRB5 in TMPM372 is invalid because there is no TB5IN pin.
Note 2: 16-bit Event Counter mode for TMRB0, TMRB4 and TMRB5 in TMPM373 is invalid because the TMPM373 doesn't have TB0IN, TB4IN and TB5IN pins.
Note 3: 16-bit Event Counter mode for TMRB0, TMRB1, TMRB4 and TMRB5 in TMPM374 is invalid because TMPM374 doesn't have TB0IN, TB1IN, TB4IN and TB5IN pin.

10.6.3 16-bit Programmable Square Wave Output Mode (PPG)

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 TB0OUT pin by triggering the timer flip-flop (TB0FF0) to reverse when the set value of the up-counter (UC) matches the set values of the timer registers (TB0RG0 and TB0RG1). Note that the set values of TB0RG0 and TB0RG1 must satisfy the following requirement:

(Set value of TB0RG0) < (Set value of TB0RG1)

Note 1: 16-bit PPG mode for TMRB5 in TMPM372 is invalid because there is no TB5OUT pin.
Note 2: 16-bit PPG mode for TMRB0, TMRB1, TMRB2, TMRB4 and TMRB5 in TMPM373 is invalid because the TMPM373 doesn't have TB0IN, TB1IN, TB2IN, TB4IN and TB5IN pins.
Note 3: 16-bit PPG mode for TMRB0, TMRB1, TMRB2, TMRB4, TMRB5 and TMRB7 in TMPM374 is invalid because TMPM374 doesn't have TB0IN, TB1IN, TB2IN, TB4IN, TB5IN and TB7IN pin.
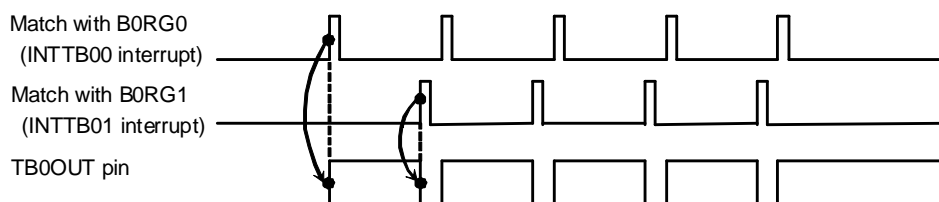
Fig. 10-2 Example of Output of Programmable Square Wave (PPG)

In this mode, by enabling the double buffering of TB0RG0 and TB0RG1, the value of register buffers are shifted into TB0RG0 and TB0RG1 when the set value of the up-counter matches the set value of TB0RG1. Since software writing time is secured by double buffer, this facilitates handling of small duties pulse.
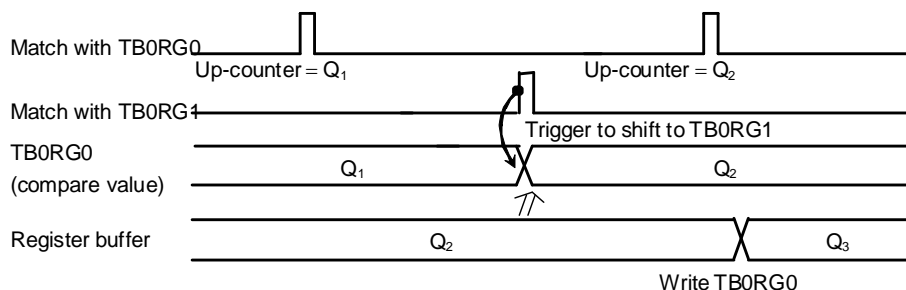
Fig. 10-3 Register Buffer Operation

The block diagram of this mode is shown below.

TB0RUN<TB0RUN>   TB0OUT (PPG output)

TB0IN0   Selector   16-bit up-counter UC0   Clear   F/F
φT1   (TB0FF0)
φT4
φT16

16-bit comparator   Match   16-bit comparator

Selector   TB0RG0   Selector   TB0RG1

TB0RG0-WR   TB0RG1-WR

TB0CR<TB0WBF>   Register buffer 0   TB0CR<TB0WBF>   Register buffer 1

Internal data bus
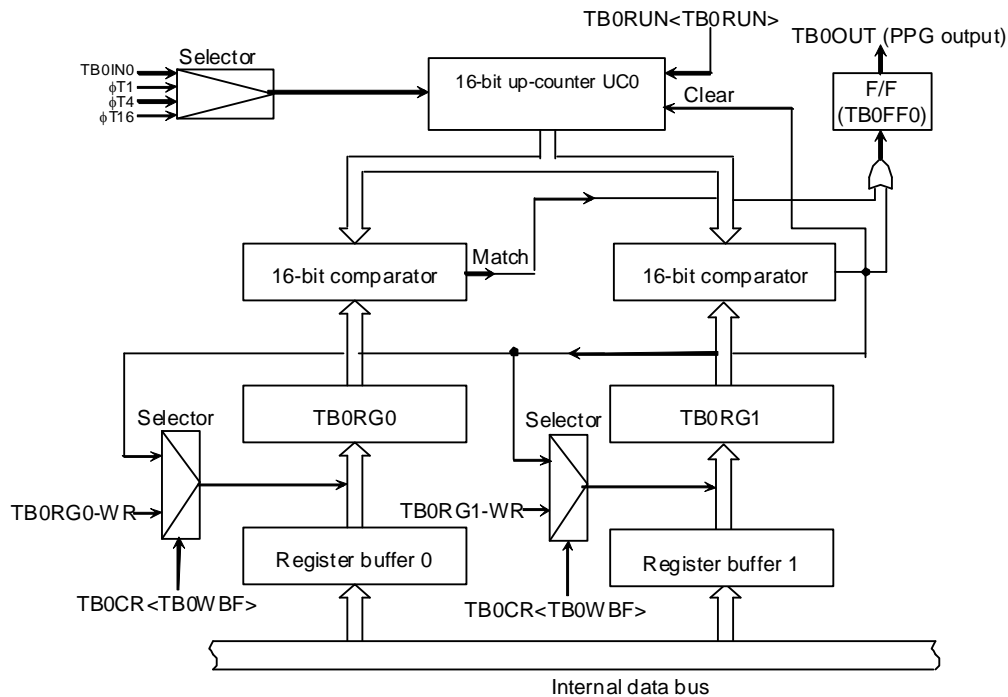
Fig. 10-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TB0EN	←	1	X	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	X	X	X	X	X	0	X	0	Stops the TMRB0
TB0RG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits *32-bits register length)
	←	*	*	*	*	*	*	*	*	
TB0RG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits *32-bits register length)
	←	*	*	*	*	*	*	*	*	
TB0CR	←	1	0	X	0	0	0	0	0	Enables the TB0RG0 double buffering. (Changes the duty/cycle when the INTTB0 interrupt is generated)
TB0FFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TB0FF0 to reverse when a match with TB0RG0 or TB0RG1 is detected, and sets the initial value of TB0FF0 to "0."
TB0MOD	←	0	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function.
						(** = 01, 10, 11)				
PACR	←	–	–	–	–	–	–	1	–	Assigns PA1 to output and TB0OUT
PAFR1	←	–	–	–	–	–	–	1	–	
TB0RUN	←	*	*	*	*	*	1	X	1	Starts TMRB0

X; Don't care –; no change

10.6.4 External trigger Programmable Square Wave Output Mode (PPG)

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

(1) The 16-bit up-counter (UC) is programmed to count up on the rising edge of the TB0IN pin (TB0CR<TRGSEL0,CSSEL0>="01"). The TB0RG0 is loaded with the pulse delay (d), and the TB0RG1 is loaded with the sum of the TB0RG0 value (d) and the pulse width (P). The above settings must be done while the 16-bit up-counter is stopped (TB0RUN<TB0RUN>=0).

(2) To enable the trigger for timer flip-flop, sets TB0FFCR<TB0E1T1, TB0E0T1> to 11. With this setting, the timer flip-flop reverses when 16-bit up-counter (UC) corresponds to TB0RG0 or TB0RG1.

(3) Sets TB0RUN<TB0RUN> to 1 to enable the count-up by an external trigger.

(4) After the generation of one-shot pulse by the external trigger, to disable reverse of the timer flip-flop or to stop 16bit counter by TB0RUN<TB0RUN> setting.

Note 1: External trigger PPG mode for TMRB5 in TMPM372 is invalid because there is no TB5OUT pin.
Note 2: External trigger PPG mode for TMRB0, TMRB1, TMRB2, TMRB4 and TMRB5 in TMPM373 is invalid because the TMPM373 doesn't have TB0IN, TB1IN, TB2IN, TB4IN and TB5IN pins.
Note 3: External trigger PPG mode for TMRB0, TMRB1, TMRB2, TMRB4, TMRB5 and TMRB7 in TMPM374 is invalid because TMPM374 doesn't have TB0IN, TB1IN, TB2IN, TB4IN, TB5IN and TB7IN pin.

Figure 10-5 shows one-shot pulse generation, with annotations showing (d) and (p).
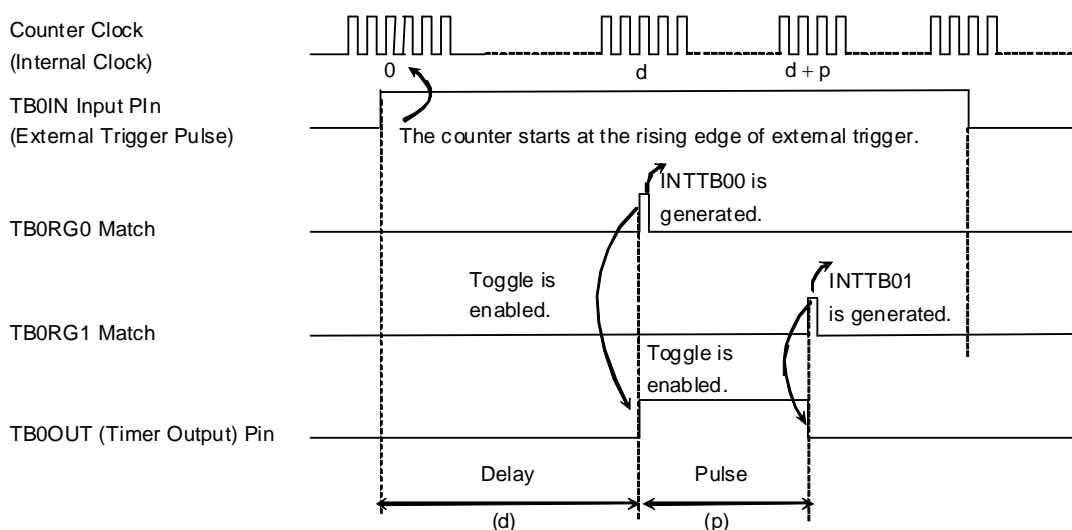
Fig. 10-5 One-shot pulse generation using an external count start trigger (with a delay)

10.7 Application example using the Capture Function

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

① One-shot pulse output triggered by an external pulse

② Pulse width measurement

> Note 1: In the TMPM372, these application examples are available except TMRB5.
> Note 2: In the TMPM373, these application examples are available only for TMRB1, TMRB2, TMRB3, TMRB6 and TMRB7.
> Note 3: In the TMPM374, these application examples are available only for TMRB2, TMRB3, TMRB6 and TMRB7.

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 TB0IN 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 (TB0CP0).

The CPU must be programmed so that an interrupt INTCAP00 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TB0RG0) to the sum of the TB0CP0 value (c) and the delay time (d), (c + d), and set the timer registers (TB0RG1) to the sum of the TB0RG0 values and the pulse width (p) of one-shot pulse, (c + d + p).

TB0RG1 change must be completed before the next match.

In addition, the timer flip-flop control registers (TB0FFCR<TB0E1T1, TB0E0T1>) must be set to "11." This enables triggering the timer flip-flop (TB0FF0) to reverse when UC matches TB0RG0 and TB0RG1. This trigger is disabled by the INTTB01 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 Fig. 10-6 One-shot Pulse Output (With Delay)."
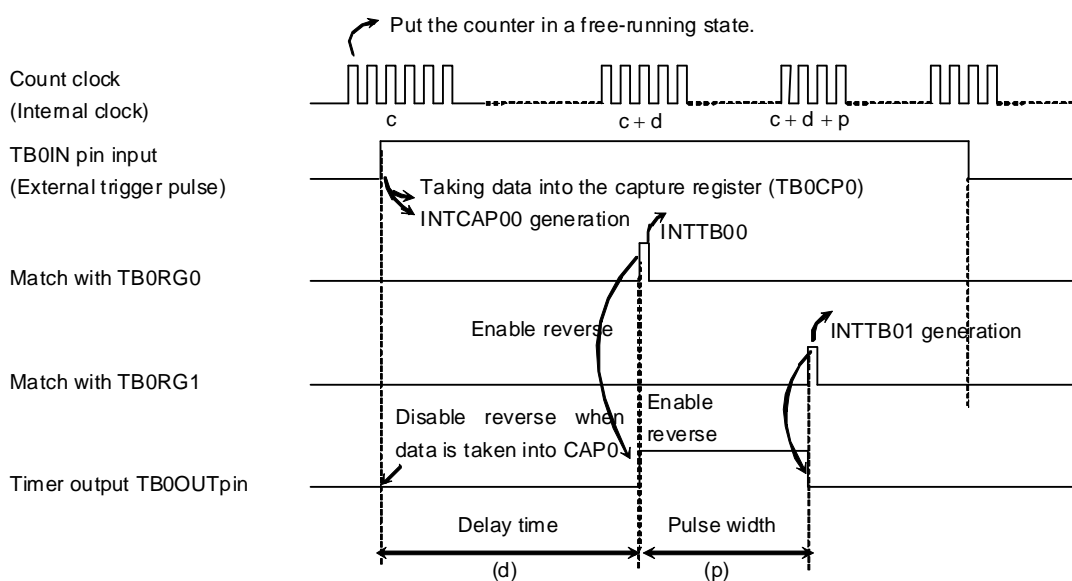
Fig. 10-6 One-shot Pulse Output (With Delay)

If a delay is not required, TB0FF0 is reversed when data is taken into TB0CP0, and TB0RG1 is set to the sum of the TB0CPO value (c) and the one-shot pulse width (p), (c + p), by generating the INTTB00 interrupt. TB0RG1 change must be completed before the next match. TB0FF0 is enabled to reverse when UC matches with TB0RG1, and is disabled by generating the INTTB01 interrupt.
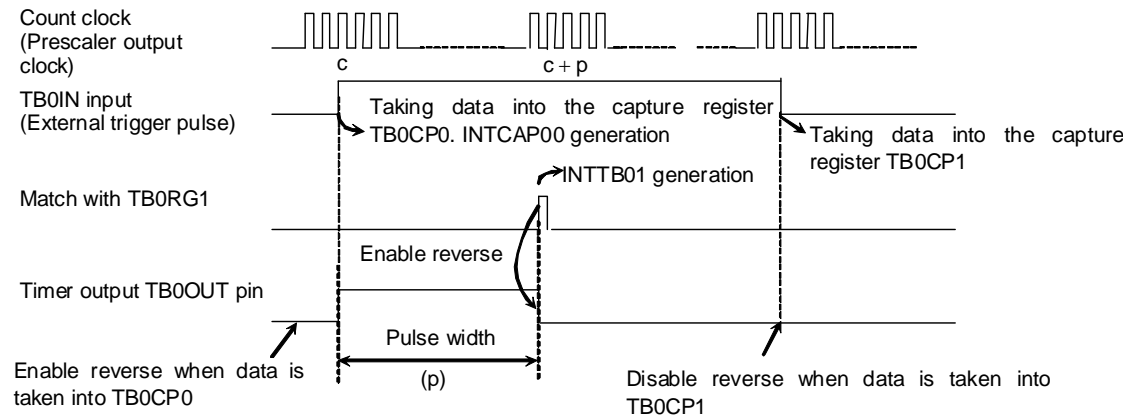
Fig. 10-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

② Pulse width measurement

By using the capture function, the "H" 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 TB0IN 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 (TB0CP0, TB0CP1). The CPU must be programmed so that INTCAP01 is generated at the falling edge of an external pulse input through the TB0IN pin.

The "H" level pulse width can be calculated by multiplying the difference between TB0CP0 and TB0CP1 by the clock cycle of an internal clock.

For example, if the difference between TB0CP0 and TB0CP1 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.
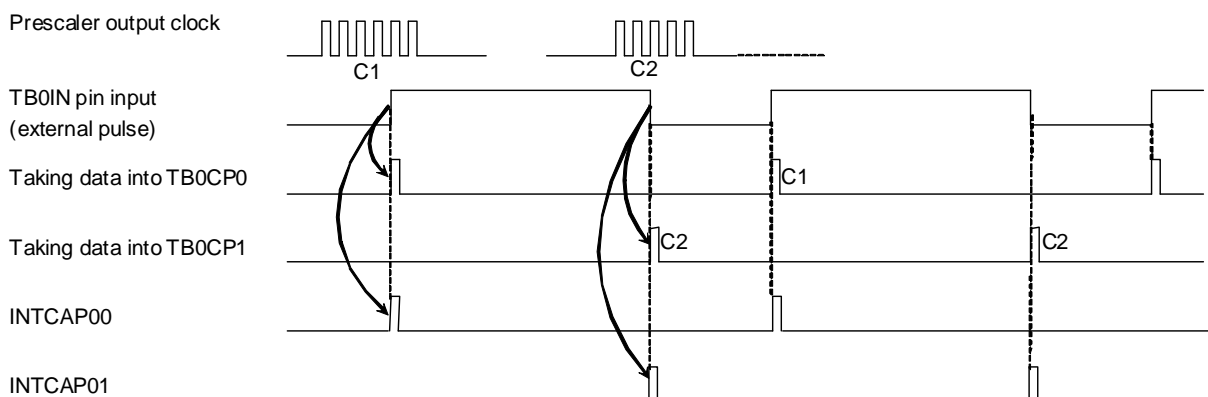
Fig. 10-8 Pulse Width Measurement

The "L" 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 INTCAP00 interrupt processing as shown in "Fig. 10-8 Pulse Width Measurement" and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

Table 11-1 Difference in the Specifications of SIO Modules

		Channel 0	Channel 1	Channel 2 (Note)	Channel 3
Pin name		TXD0 (PE0) RXD0 (PE1) $\overline{CTS0}$ /SCLK0 (PE2)	TXD1 (PA5) RXD1 (PA6) $\overline{CTS1}$ /SCLK1 (PA4)	TXD2 (PD5) RXD2 (PD6) $\overline{CTS2}$ /SCLK2 (PD4)	TXD3 (PF3) RXD3 (PF4) $\overline{CTS3}$/SCLK3 (PF2)
Interrupt		INTRX0 INTTX0	INTRX1 INTTX1	INTRX2 INTTX2	INTRX3 INTTX3
In the UART mode, TMRB output to use for the serial transfer clock		TB4OUT	TB4OUT	TB7OUT	TB7OUT
Register name (address)	Enable register	SC0EN 0x4002_0080	SC1EN 0x4002_00C0	SC2EN 0x4002_0100	SC3EN 0x4002_0140
	Transmit/ receive buffer register	SC0BUF 0x4002_0084	SC1BUF 0x4002_00C4	SC2BUF 0x4002_0104	SC3BUF 0x4002_0144
	Control register	SC0CR 0x4002_0088	SC1CR 0x4002_00C8	SC2CR 0x4002_0108	SC3CR 0x4002_0148
	Mode control register 0	SC0MOD0 0x4002_008C	SC1MOD0 0x4002_00CC	SC2MOD0 0x4002_010C	SC3MOD0 0x4002_014C
	Baud rate generator control	SC0BRCR 0x4002_0090	SC1BRCR 0x4002_00D0	SC2BRCR 0x4002_0110	SC3BRCR 0x4002_0150
	Baud rate generator control 2	SC0BRADD 0x4002_0094	SC1BRADD 0x4002_00D4	SC2BRADD 0x4002_0114	SC3BRADD 0x4002_0154
	Mode control register 1	SC0MOD1 0x4002_0098	SC1MOD1 0x4002_00D8	SC2MOD1 0x4002_0118	SC3MOD1 0x4002_0158
	Mode control register 2	SC0MOD2 0x4002_009C	SC1MOD2 0x4002_00DC	SC2MOD2 0x4002_011C	SC3MOD2 0x4002_015C
	Receive FIFO configuration register	SC0RFC 0x4002_00A0	SC1RFC 0x4002_00E0	SC2RFC 0x4002_0120	SC3RFC 0x4002_0160
	Transmit FIFO configuration register	SC0TFC 0x4002_00A4	SC1TFC 0x4002_00E4	SC2TFC 0x4002_0124	SC3TFC 0x4002_0164
	Receive FIFO status register	SC0RST 0x4002_00A8	SC1RST 0x4002_00E8	SC2RST 0x4002_0128	SC3RST 0x4002_0168
	Transmit FIFO status register	SC0TST 0x4002_00AC	SC1TST 0x4002_00EC	SC2TST 0x4002_012C	SC3TST 0x4002_016C
	FIFO configuration register	SC0FCNF 0x4002_00B0	SC1FCNF 0x4002_00F0	SC2FCNF 0x4002_0130	SC3FCNF 0x4002_0170

Note: SIO2 is not available in TMPM373/374. Do not access registers for SIO2 in TMPM373/374.

- Mode 0 (I/O Interface mode) /LSB first

| bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

← Transmission direction

- Mode 0 (I/O Interface mode) /MSB first

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

← Transmission direction

- Mode 1 (7 bits UART mode)

Without parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | stop

With parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | parity | stop

- Mode 2 (8 bits UART mode)

Without parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | stop

With parity | start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | parity | stop

- Mode 3 (9 bits UART mode)

| start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | stop

| start | bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | bit 8 | Stop(Wake-up)

IIf bit 8=1, represents address (select code)
IIf bit 8=0, represents data.

Fig. 11-1 Data Format

11.2 Block Diagram (Channel 0)
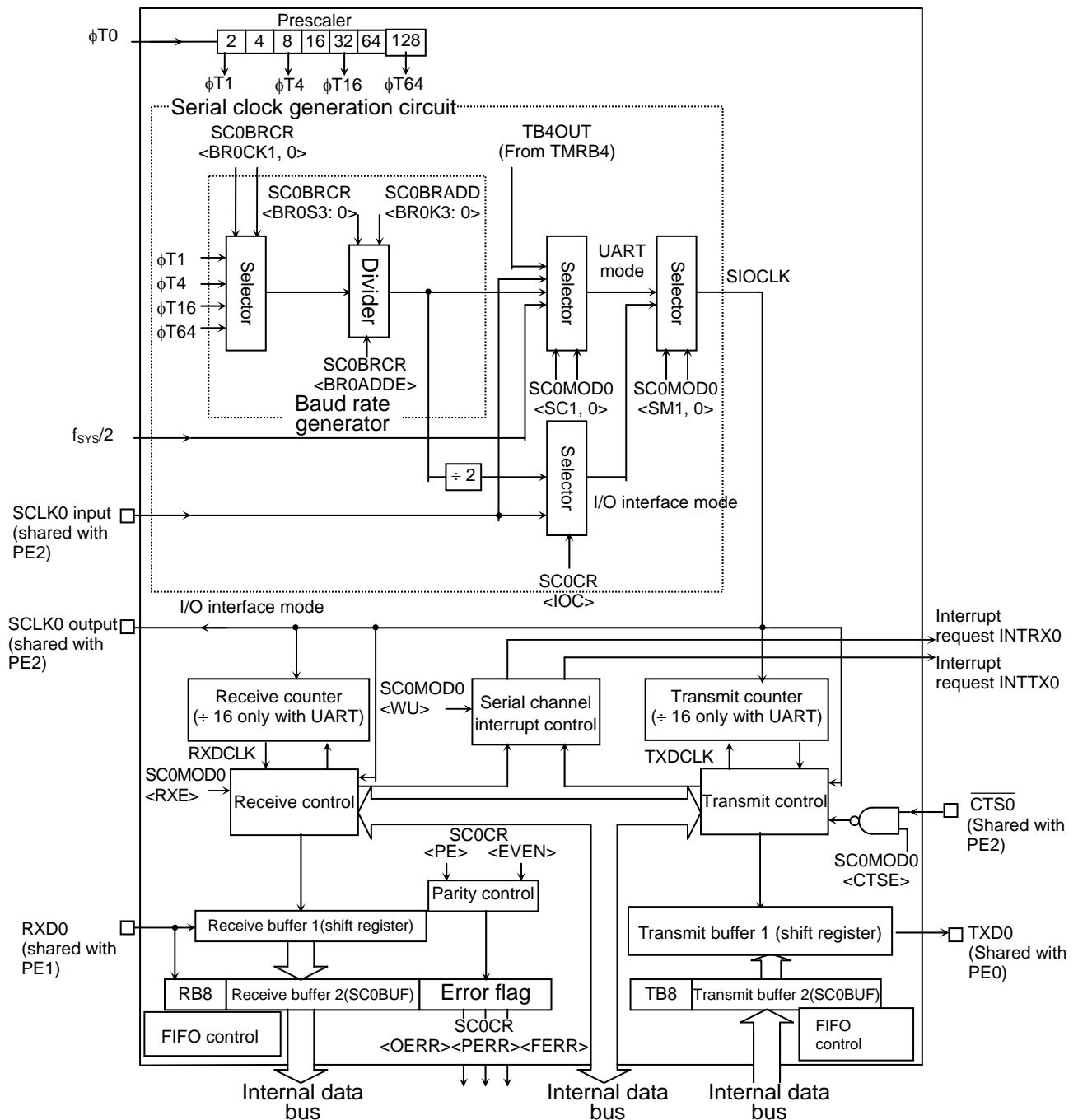
Fig. 11-2 SIO0 Block Diagram

11.3 Operation of Each Circuit (Channel 0)

11.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock ϕT0 to the prescaler is selected by SYSCR1 of CG <PRCK2:0> to provide the frequency of either fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32.

The clock frequency fperiph is either the clock "fgear," to be selected by SYSCR1<FPSEL> of CG, or the clock "fc" before it is divided by the clock gear.

The prescaler becomes active only when the baud rate generator is selected for generating the serial transfer clock. Table 11-2 list the prescaler output clock resolution.

Table 11-2 Clock Resolution to the Baud Rate Generator   @ = 80MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution			
			ϕT1	ϕT4	ϕT16	ϕT64
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025μs)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)
		001 (fperiph/2)	$fc/2^2$ (0.05μs)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)
		010 (fperiph/4)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		011 (fperiph/8)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		100 (fperiph/16)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		101 (fperiph/32)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
	100(fc/2)	000 (fperiph/1)	$fc/2^2$ (0.05μs)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)
		001 (fperiph/2)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		010 (fperiph/4)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		011 (fperiph/8)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		100 (fperiph/16)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
		101 (fperiph/32)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)	$fc/2^{13}$ (102.4μs)
	101(fc/4)	000 (fperiph/1)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		001 (fperiph/2)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		010 (fperiph/4)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		011 (fperiph/8)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
		100 (fperiph/16)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)	$fc/2^{13}$ (102.4μs)
		101 (fperiph/32)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)	$fc/2^{14}$ (204.8μs)
	110(fc/8)	000 (fperiph/1)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		001 (fperiph/2)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		010 (fperiph/4)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
		011 (fperiph/8)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)	$fc/2^{13}$ (102.4μs)
		100 (fperiph/16)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)	$fc/2^{14}$ (204.8μs)
		101 (fperiph/32)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)	$fc/2^{13}$ (102.4μs)	$fc/2^{15}$ (409.6μs)
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025μs)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)
		001 (fperiph/2)	$fc/2^2$ (0.05μs)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)
		010 (fperiph/4)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		011 (fperiph/8)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		100 (fperiph/16)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		101 (fperiph/32)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
	100(fc/2)	000 (fperiph/1)	—	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)
		001 (fperiph/2)	$fc/2^2$ (0.05μs)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)
		010 (fperiph/4)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		011 (fperiph/8)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		100 (fperiph/16)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		101 (fperiph/32)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
	101(fc/4)	000 (fperiph/1)	—	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)
		001 (fperiph/2)	—	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)
		010 (fperiph/4)	$fc/2^3$ (0.1μs)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		011 (fperiph/8)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		100 (fperiph/16)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		101 (fperiph/32)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)
	110(fc/8)	000 (fperiph/1)	—	—	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)
		001 (fperiph/2)	—	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)
		010 (fperiph/4)	—	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)
		011 (fperiph/8)	$fc/2^4$ (0.2μs)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)
		100 (fperiph/16)	$fc/2^5$ (0.4μs)	$fc/2^7$ (1.6μs)	$fc/2^9$ (6.4μs)	$fc/2^{11}$ (25.6μs)
		101 (fperiph/32)	$fc/2^6$ (0.8μs)	$fc/2^8$ (3.2μs)	$fc/2^{10}$ (12.8μs)	$fc/2^{12}$ (51.2μs)

> **(Note 1) The prescaler output clock** ϕTn **must be selected so that the relationship "**ϕTn < fsys**"**
>   **is satisfied (so that** ϕTn **is slower than fsys).**
> **(Note 2) Do not change the clock gear while SIO is operating.**
> **(Note 3) The horizontal lines in the above table indicate that the setting is prohibited.**

The serial interface baud rate generator uses four different clocks, i.e., ϕT1, ϕT4, ϕT16 and ϕT64, supplied from the prescaler output clock.

11.3.2 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses either the ϕT1, ϕT4, ϕT16 or ϕT64 clock supplied from the 7-bit prescaler. This input clock selection is made by setting the baud rate generator control register, SC0BRCR <BR0CK1:0>.

The baud rate generator contains built-in dividers for divide by 1, N + m/16 (N=2~15, m=1~15), and 16. The division is performed according to the settings of the baud rate generator control registers SC0BRCR <BR0ADDE> <BR0S3:0> and SC0BRADD <BR0K3:0> to determine the resulting transfer rate.

The highest baud rate of each mode is limited.

- UART mode

    1) If SC0BRCR <BR0ADDE> = 0,

       The setting of SC0BRADD <BR0K3:0> is ignored and the counter is divided by N where N is the value set to SC0BRCR <BR0S3:0>. (N = 1 to 16).

    2) If SC0BRCR <BR0ADDE> = 1,

       The N + (16 - K)/16 division function is enabled and the division is made by using the values N (set in SC0BRCR <BR0S3:0>) and K (set in SC0BRADD<BR0K3:0>). (N = 2 to 15, K = 1 to 15)

       > **(Note) For the N values of 1 and 16, the above N+(16-K)/16 division function is inhibited. So, be sure to set SC0BRCR<BR0ADDE> to "0."**

- I/O interface mode

    The N + (16 - K)/16 division function cannot be used in the I/O interface mode. Be sure to divide by N, by setting SC0BRCR <BR0ADDE> to "0".

- Baud rate calculation to use the baud rate generator:

    1) UART mode

       $$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} \,/16$$

       The highest baud rate out of the baud rate generator is 2.5 Mbps.

       The fsys frequency, which is independent of the baud rate generator, can be used as the serial clock. In this case, the highest baud rate will be 5.0 Mbps.

    2) I/O interface mode

       $$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} \,/2$$

       When it uses a double buffer, the highest baud rate generated with the baud rate generator becomes 20Mbps. (If double buffering is not used, the highest baud rate will be 10 Mbps).

• <u>Example baud rate setting:</u>

1) Division by an integer (divide by N):

Selecting fc = 78.642 MHz for fperiph, setting ϕT0 to fperiph/16, using the baud rate generator input clock ϕT1, setting the divide ratio N (SC0BRCR<BR0S3:0>) = 4, and setting SC0BRCR<BR0ADDE> = "0," the resulting baud rate in the UART mode is calculated as follows:

* Clocking conditions   System clock   :   High-speed (fc)

High speed clock gear :   x 1 (fc)

Prescaler clock   : f_{periph}/16 (f_{periph} = f_{sys})

Baud rate = $\dfrac{fc/32}{4}$ /16

= $78.642 \times 10^6 \div 32 \div 4 \div 16 \fallingdotseq 38400$ (bps)

(Note)   The divide by (N + (16-K)/16) function is inhibited and thus SC0BRADD <BR0K3:0> is ignored.

---

2) For divide by N + (16-K)/16 (only for UART mode):

Selecting fc = 9.6 MHz for fperiph, setting ϕT0 to fperiph/8, using the baud rate generator input clock ϕT1, setting the divide ratio N (SC0BRCR<BR0S3:0>) = 7, setting K (SC0BRADD<BR0K3:0>) = 3, and selecting SC0BRCR<BR0ADDE> = 1, the resulting baud rate is calculated as follows:

* Clocking conditions $\Big\{$ System clock   :   High-speed (fc)

High-speed clock gear :   x 1 (fc)

Prescaler clock   : f_{periph}/4 (f_{periph} = f_{sys})

Baud rate = $\dfrac{fc/16}{7 + \dfrac{(16 - 3)}{16}}$ /16

= $9.6 \times 10^6 \div 16 \div (7 + \dfrac{13}{16}) \div 16 = 4800$ (bps)

Also, an external clock input may be used as the serial clock. The resulting baud rate calculation is shown below:

- Baud rate calculation for an external clock input:

    1) UART mode

    Baud Rate = external clock input / 16

    In this, the period of the external clock input must be equal to or greater than 2/fsys.

    The highest baud rate must be less than $80 \div 2 \div 16 = 2.5$ Mbps.

    2) I/O interface mode

    Baud Rate = external clock input

    When double buffering is used, it is necessary to satisfy the following relationship:

    External clock input period > 6/fsys

    The highest baud rate must be less than $80 \div 6 = 13.3$ Mbps.

    When double buffering is not used, it is necessary to satisfy the following relationship:

    External clock input period > 8/fsys

    The highest baud rate must be less than $80 \div 8 = 10$ Mbps.

    The baud rate examples for the UART mode are shown in Table 11-3 and Table 11-4.

Table 11-3 Selection of UART Baud Rate

(Using the baud rate generator with SC0BRCR <BR0ADDE> = 0)   Unit: (kbps)

fc [MHz]	Divide ratio N (Set to SC0BRCR <BR0S3 : 0>) Input clock	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

(Note) **This table shows the case where the system clock is set to fc, the clock gear is set to fc/1, and the prescaler clock is set to $f_{periph}/2$.**

Table 11-4 Selection of UART Baud Rate

(The TMRB4 timer output (internal TB4OUT) is used with the timer input clock set to φT1.)

Unit: (Kbbs)

TB4RG	fc 80 MHz	9.8304 MHz	8 MHz
0x0001	625.0	76.8	62.5
0x0002	312.5	38.4	31.25
0x0003		25.6	
0x0004	156.25	19.2	15.625
0x0005	125.0	15.36	12.5
0x0006		12.8	
0x0008	78.125	9.6	
0x000A	62.5	7.68	6.25
0x0010	39.0625	4.8	
0x0014	31.25	3.84	3.125

Baud rate calculation to use the TMRB4 timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by SYSCR0<PRCK2:0>}}{TB4RG \times 2 \times \underline{2} \times 16}$$

└─ (When input clock to the timer TMRB4 is φT1)

(Note 1) **In the I/O interface mode, the TMRB4 timer output signal cannot be used internally as the transfer clock.**

(Note 2) **This table shows the case where the system clock is set to fc, the clock gear is set to fc, and the prescaler clock is set to $f_{periph}/2$.**

11.3.3 Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

- I/O interface mode

    In the SCLK output mode with the SC0CR <IOC> serial control register set to "0," the output of the previously mentioned baud rate generator is divided by 2 to generate the basic clock.

    In the SCLK input mode with SC0CR <IOC> set to "1," rising and falling edges are detected according to the SC0CR <SCLKS> setting to generate the basic clock.

- Asynchronous (UART) mode :

    According to the settings of the serial control mode register SC0MOD0 <SC1:0>, either the clock from the baud rate register, the system clock (f_{SYS}), the internal output signal of the TMRB4 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

11.3.4 Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by SIOCLK. Sixteen SIOCLK clock pulses are used in receiving a single data bit while 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.

11.3.5 Receive Control Unit

- I/O interface mode:

    In the SCLK output mode with SC0CR <IOC> set to "0," the RXD0 pin is sampled on the falling edge of the shift clock output to the SCLK0 pin.

    In the SCLK input mode with SC0CR <IOC> set to "1," the serial receive data RXD0 pin is sampled on the rising or falling edge of SCLK input depending on the SC0CR <SCLKS> setting.

- Asynchronous (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.

11.3.6 Receive Buffer

The receive buffer is of a dual structure to prevent overrun errors. The first receive buffer (a shift register) stores the received data bit-by-bit. When a complete set of bits have been stored, they are moved to the second receive buffer (SC0BUF). At the same time, the receive buffer full flag (SC0MOD2 "RBFLL") is set to "1" to indicate that valid data is stored in the second receive buffer. However, if the receive FIFO is set enabled, the receive data is moved to the receive FIFO and this flag is immediately cleared.

If the receive FIFO has been disabled (SCOFCNF <CNFG> = 0 and SC0MOD1<FDPX1:0> =01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1<FDPX1:0> = 01), an interrupt will be generated according to the SC0RFC <RIL1:0> setting.

The CPU will read the data from either the second receive buffer (SC0BUF) or from the receive FIFO (the address is the same as that of the receive buffer). If the receive FIFO has not been enabled, the receive buffer full flag <RBFLL> is cleared to "0" by the read operation. The next data received can be stored in the first receive buffer even if the CPU has not read the previous data from the second receive buffer (SC0BUF) or the receive FIFO.

If SCLK is set to generate clock output in the I/O interface mode, the double buffer control bit SC0MOD2 <WBUF> can be programmed to enable or disable the operation of the second receive buffer (SCOBUF).

By disabling the second receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (SCOFCNF <CNFG> = 0 and <FDPX1:0> = 01), handshaking with the other side of communication can be enabled and the SCLK output stops each time one frame of data is transferred. In this setting, the CPU reads data from the first receive buffer. By the read operation of CPU, the SCLK output resumes.

If the second receive buffer (i.e., double buffering) is enabled but the receive FIFO is not enabled, the SCLK output is stopped when the first receive data is moved from the first receive buffer to the second receive buffer and the next data is stored in the first buffer filling both buffers with valid data. When the second receive buffer is read, the data of the first receive buffer is moved to the second receive buffer and the SCLK output is resumed upon generation of the receive interrupt INTRX0. Therefore, no buffer overrun error will be caused in the I/O interface SCLK output mode regardless of the setting of the double buffer control bit SC0MOD2 <WBUF>.

If the second receive buffer (double buffering) is enabled and the receive FIFO is also enabled (SCNFCNF <CNFG> = 1 and <FDPX1:0> = 01/11), the SCLK output will be stopped when the receive FIFO is full (according to the setting of SCOFNCF <RFST>) and both the first and second receive buffers contain valid data. Also in this case, if SCOFCNF <RXTXCNT> has been set to "1," the receive control bit RXE will be automatically cleared upon suspension of the SCLK output. If it is set to "0," automatic clearing will not be performed.

> **(Note) In this mode, the SC0CR <OEER> flag is insignificant and the operation is undefined. Therefore, before switching from the SCLK output mode to another mode, the SC0CR register must be read to initialize this flag.**

In other operating modes, the operation of the second receive buffer is always valid, thus improving the performance of continuous data transfer. If the receive FIFO is not enabled, an overrun error occurs when the data in the second receive buffer (SC0BUF) has not been read before the first receive buffer is full with the next receive data. If an overrun error occurs, data in the first receive buffer will be lost while data in the second receive buffer and the contents of SC0CR <RB8> remain intact. If the receive FIFO is enabled, the FIFO must be read before the FIFO is full and the second receive buffer is written by the next data through the first buffer. Otherwise, an overrun error will be generated and the receive FIFO overrun error flag will be set. Even in this case, the data already in the receive FIFO remains intact.

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 SC0CR <RB8>.

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by

setting the wake-up function SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1."

11.3.7 Receive FIFO Buffer

In addition to the double buffer function already described, data may be stored using the receive FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte(maximum) receive buffer can be enabled. Also, in the UART mode or I/O interface mode, data may be stored up to a predefined fill level. When the receive FIFO buffer is to be used, be sure to enable the double buffer function.

If data with parity bit is to be received in the UART mode, parity check must be performed each time a data frame is received.

11.3.8 Receive FIFO Operation

① I/O interface mode with SCLK output:

The following example describes the case a 6-byte data stream is received in the half duplex mode:

SC0MOD1<6:5>=01: Transfer mode is set to half duplex mode.

SC0FCNF <4:0>=10111: Automatically inhibits continued 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.

SC0RFC<1:0>=00: Sets the interrupt to be generated at fill level 4.

SC0RFC<7:6>=11: Clears receive FIFO and sets the condition of interrupt generation.

In this condition, data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. When the data is stored all in the receive shift register, receive buffer and receive FIFO, SCxMOD0<RXE> is automatically cleared and the receive operation is finished.
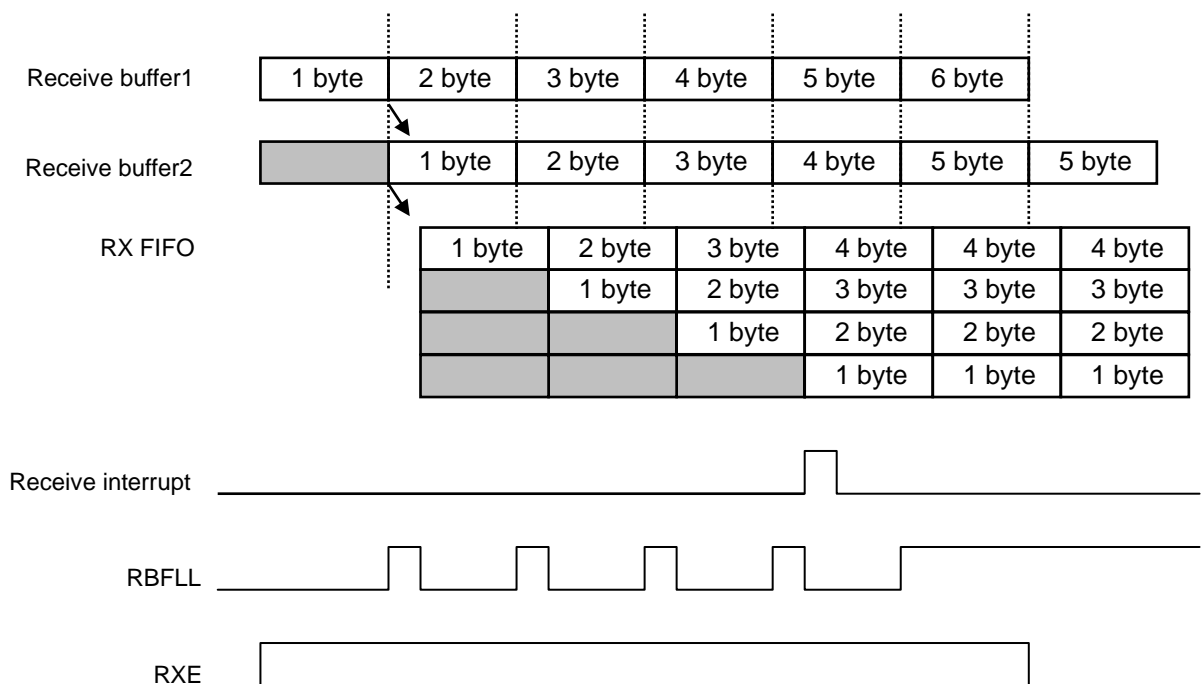

Fig. 11-3 Receive FIFO Operation

Serial Channel

② I/O interface mode with SCLK input:

The following example describes the case a 6-byte data stream is received:

SC0MOD1<6:5>=01: Transfer mode is set to half duplex mode.

SC0FCNF <1:0> = 10101: Automatically allows continued reception after reaching the fill level.
SC0RFC <1:0> = 00: Sets the interrupt to be generated at fill level 4.

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

The number of bytes to be used in the receive FIFO is the maximum allowable number.

In this condition, 6-byte data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. When the data is stored all in the receive shift register, receive buffer and receive FIFO, receive FIFO interrupt is generated. This setting enables the next data reception as well. Data can be received continuously by reading data in FIFO according to the input clock.

Receive buffer1	1 byte	2 byte	3 byte	4 byte	5 byte	6 byte	

Receive buffer2		1 byte	2 byte	3 byte	4 byte	5 byte	5 byte

RX FIFO		1 byte	2 byte	3 byte	4 byte	4 byte	4 byte
			1 byte	2 byte	3 byte	3 byte	3 byte
				1 byte	2 byte	2 byte	2 byte
					1 byte	1 byte	1 byte

Receive interrupt

RBFLL

RXE
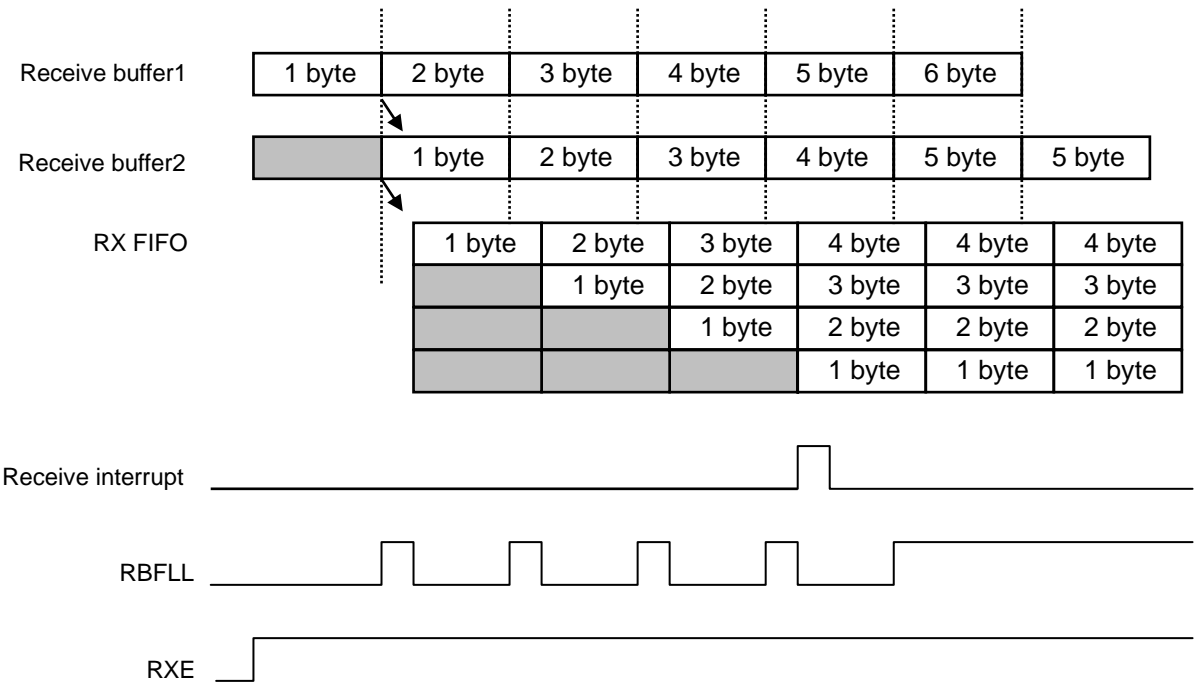
Fig. 11-4 Receive FIFO Operation

11.3.9 Transmit Counter

The transmit counter is a 4-bit binary counter and is counted by SIOCLK as in the case of the receive counter. In UART mode, it generates a transmit clock (TXDCLK) on every 16th clock pulse.
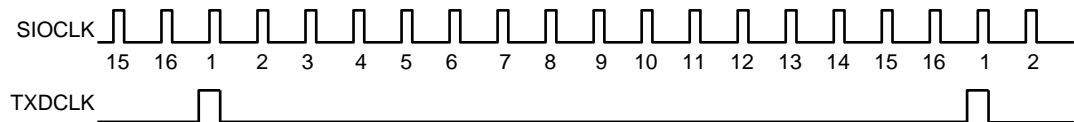

Fig. 11-5 Transmit Clock Generation

11.3.10   Transmit Control Unit

- I/O interface mode:

    In the SCLK output mode with SC0CR <IOC> set to "0," each bit of data in the transmit buffer is output to the TXD0 pin on the falling edge of the shift clock output from the SCLK0 pin.

    In the SCLK input mode with SC0CR <IOC> set to "1," each bit of data in the transmit buffer is output to the TXD0 pin on the rising or falling edge of the input SCLK signal according to the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

    When the CPU writes data to the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock (TXDSFT) is also generated.

- Handshake function

    The $\overline{CTS}$ pin enables frame by frame data transmission so that overrun errors can be prevented. This function can be enabled or disabled by SC0MOD0 <CTSE>.

    When the $\overline{CTS0}$ pin is set to the "H" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{CTS0}$ pin returns to the "L" level. However in this case, the INTTX0 interrupt is generated, the next transmit data is requested to the CPU, data is written to the transmit buffer, and it waits until it is ready to transmit data.

    Although no $\overline{RTS}$ pin is provided, a handshake control function can be easily implemented by assigning a port for the $\overline{RTS}$ function. By setting the port to "H" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.
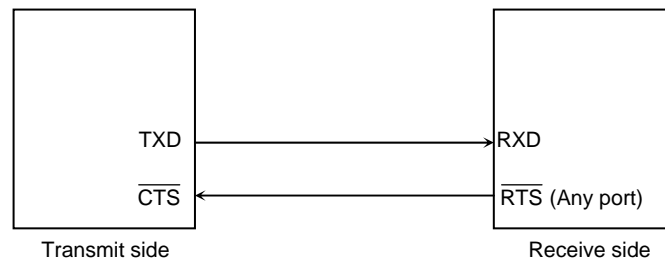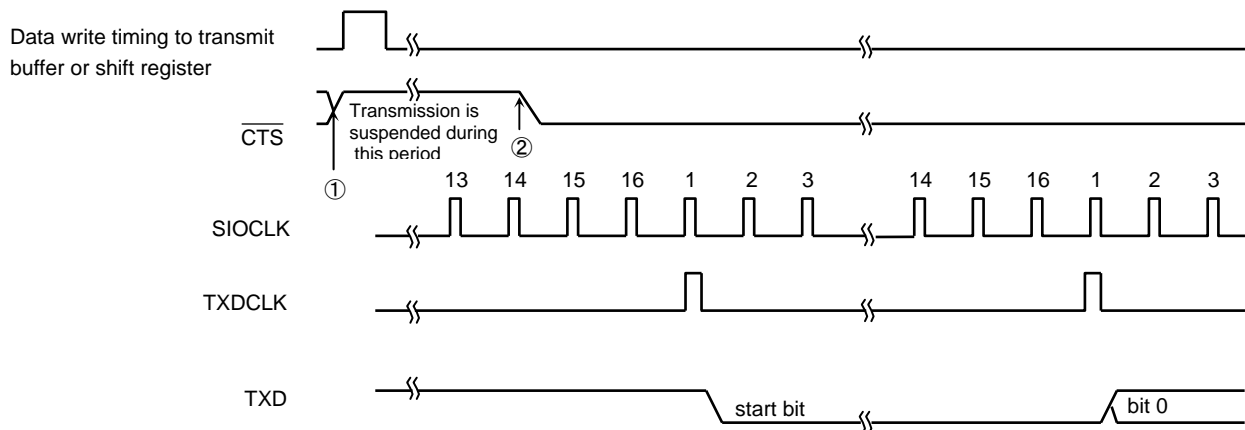

Fig. 11-6 Handshake Function

Fig. 11-7 $\overline{CTS}$ (Clear to Transmit) Signal Timing

(Note 1) **If the $\overline{CTS}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.**

(Note 2) **Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{CTS}$ is set to "L."**

11.3.11 Transmit Buffer

The transmit buffer (SC0BUF) is in a dual structure. The double buffering function may be enabled or disabled by setting the double buffer control bit <WBUF> in serial mode control register 2 (SC0MOD2). If double buffering is enabled, data written to Transmit Buffer 2 (SCOBUF) is moved to Transmit Buffer 1 (shift register).

If the transmit FIFO has been disabled (SCOFCNF <CNFG> = 0 or 1 and SC0MOD1＜FDPX1:0＞＝01), the INTTX0 interrupt is generated at the same time and the transmit buffer empty flag <TBEMP> of SC0MOD2 is set to "1." This flag indicates that Transmit Buffer 2 is now empty and that the next transmit data can be written. When the next data is written to Transmit Buffer 2, the <TBEMP> flag is cleared to "0."

If the transmit FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1＜FDPX1:0＞＝10/11), any data in the transmit FIFO is moved to the Transmit Buffer 2 and <TBEMP> flag is immediately cleared to "0." The CPU writes data to Transmit Buffer 2 or to the transmit FIFO.

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in Transmit Buffer 2 before the next frame clock input, which occurs upon completion of data transmission from Transmit Buffer 1, an under-run error occurs and a serial control register (SC0CR) <PERR> parity/under-run flag is set.

If the transmit FIFO is enabled in the I/O interface SCLK input mode, when data transmission from Transmit Buffer 1 is completed, the Transmit Buffer 2 data is moved to Transmit Buffer 1 and any data in transmit FIFO is moved to Transmit Buffer 2 at the same time.

If the transmit FIFO is disabled in the I/O interface SCLK output mode, when data in Transmit Buffer 2 is moved to Transmit Buffer 1 and the data transmission is completed, the SCLK output stops. So, no under-run errors can be generated.

If the transmit FIFO is enabled in the I/O interface SCLK output mode, the SCLK output stops upon completion of data transmission from Transmit Buffer 1 if there is no valid data in the transmit FIFO.

Note) In the I/O interface SCLK output mode, the SC0CR <PEER> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the SCLK output mode to another mode, SC0CR must be read in advance to initialize the flag.

If double buffering is disabled, the CPU writes data only to Transmit Buffer 1 and the transmit interrupt INTTX0 is generated upon completion of data transmission.

If handshaking with the other side is necessary, set the double buffer control bit <WBUF> to "0" (disable) to disable Transmit Buffer 2; any setting for the transmit FIFO should not be performed.

11.3.12   Transmit FIFO Buffer

In addition to the double buffer function already described, data may be stored using the transmit FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte transmit buffer can be enabled. In the UART mode or I/O interface mode, up to 4 bytes of data may be stored.
If data is to be transmitted with a parity bit in the UART mode, parity check must be performed on the receive side each time a data frame is received.

> **Note)   Please clear transmission FIFO after setting of the forwarding mode of SIO (half duplex/full duplex) and permission (SC0FCNF<CNFG>="1") of FIFO when you use transmission FIFO Buffer.**

11.3.13   Transmit FIFO Operation

①   I/O interface mode with SCLK output (normal mode):

SC0MOD1<6:5>=10: Transfer mode is set to half duplex mode.

SC0FCNF <4:0> = 01011: Inhibits continued transmission after reaching the fill level.

SC0TFC <1:0> = 00: Sets the interrupt to be generated at fill level 0.

SC0TFC <7:6> =11: Clears transmit FIFO and sets the condition of interrupt generation

In this condition, data transmission can be initiated by setting the transfer mode to half duplex, writing 5 bytes of data to the transmit buffer and transmit FIFO , and setting the <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.
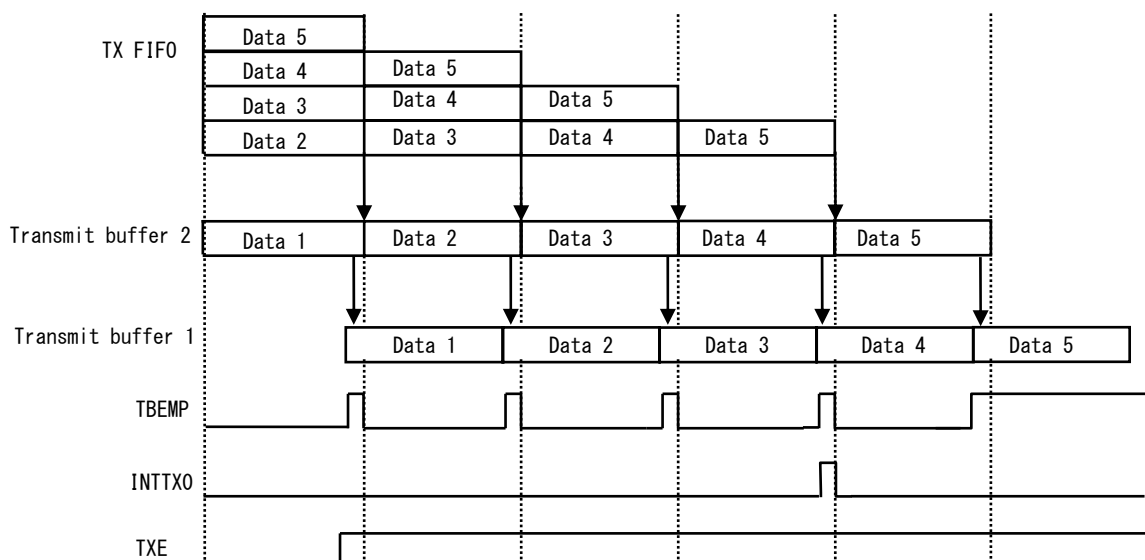

Fig. 11-8 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

SC0MOD1<6:5>=10: Transfer mode is set to half duplex mode.

SC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

SC0TFC <1:0> = 00: Sets the interrupt to be generated at fill level 0.

SC0TFC <7:6> = 11: Clears the receive FIFO and sets the condition of interrupt
generation.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 5 bytes of data to the transmit buffer and the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated.
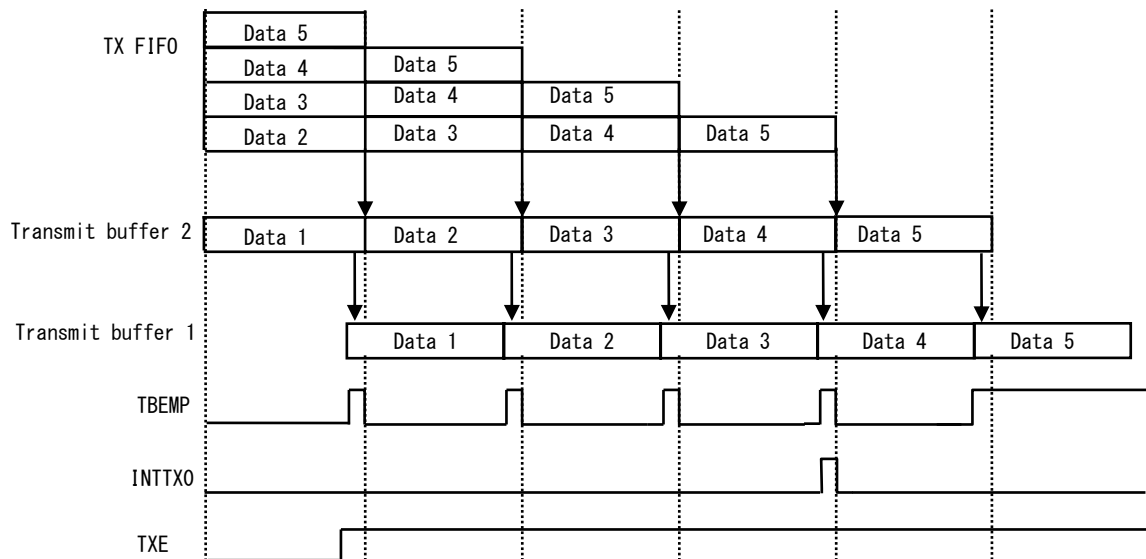

Fig. 11-9 Transmit FIFO Operation

11.3.14 Parity Control Circuit

If the parity addition bit <PE> of the serial control register SC0CR is set to "1," data is sent with the parity bit. Note that the parity bit may be used only in the 7- or 8-bit UART mode. The <EVEN> bit of SC0CR selects either even or odd parity.

Upon data transmission, the parity control circuit automatically generates the parity with the data written to the transmit buffer (SC0BUF). After data transmission is complete, the parity bit will be stored in SC0BUF bit 7 <TB7> in the 7-bit UART mode and in bit 7 <TB8> in the serial mode control register SC0MOD in the 8-bit UART mode. The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

Upon data reception, the parity bit for the received data is automatically generated while the data is shifted to receive buffer 1 and moved to receive buffer 2 (SC0BUF). In the 7-bit UART mode, the parity generated is compared with the parity stored in SC0BUF <RB7>, while in the 8-bit UART mode, it is compared with the bit 7 <RB8> of the SC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the SC0CR register is set. When the FIFO is used, the <PERR> indicates that one or more data have the parity error among the receiving data.

In the I/O interface mode, the SC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

11.3.15 Error Flag

Three error flags are provided to inprove the reliability of received data.

1. Overrun error <OERR>: Bit 4 of the serial control register SC0CR

   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 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). This flag is set to "0" when it is read. In the I/O interface SCLK output mode, no overrun error is generated and therefore, this flag is inoperative and the operation is undefined.

   > **Note)** **Changing the mode from I/O interface SCLK output mode to the other mode, read SC0CR to clear SC0CR<OERR> flag.**

2. Parity error/under-run error <PERR>: Bit 3 of the SC0CR register

   In the UART mode, this bit is set to "1" when a parity error is generated. A parity error is generated when the parity generated from the received data is different from the parity received. This flag is set to "0" when it is read.

   In the I/O interface mode, this bit indicates an under-run error. When the double buffer control bit <WBUF> of the serial mode control register SC0MOD2 is set to "1" in the SCLK input mode, if no data is set to the transmit double buffer before the next data transfer clock after completing the transmission from the transmit shift register, this error flag is set to "1" indicating an under-run error. If the transmit FIFO is enabled, any data content in the transmit FIFO will be moved to the buffer. When the transmit FIFO and the double buffer are both empty, an under-run error will be generated. Because no under-run errors can be generated in the SCLK output mode, this flag is inoperative

and the operation is undefined. If Transmit Buffer 2 is disabled, the under-run flag
<PERR> will not be set. This flag is set to "0" when it is read.

> **Note)** **Changing the mode from I/O interface SCLK output mode to the other
> mode, read SC0CR to clear SC0CR<PERR> flag.**

3. Framing error <FERR>: Bit 2 of the SC0CR register

   In the UART mode, this bit is set to "1" when a framing error is generated. This flag is
   set to "0" when it is read. 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
   <SBLEN> (stop bit length) setting of the serial mode control register 2, SC0MOD2, the
   stop bit status is determined by only 1 bit on the receive side.

Operation mode	Error flag	Function
UART	OERR	Overrun error flag
	PERR	Parity error flag
	FERR	Framing error flag
I/O Interface (SCLK input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O Interface (SCLK output)	OERR	Operation undefined
	PERR	Operation undefined
	FERR	Fixed to 0

11.3.16 Direction of Data Transfer

In the I/O interface mode, the direction of data transfer can be switched between "MSB first"
and "LSB first" by the data transfer direction setting bit <DRCHG> of the SC0MOD2 serial
mode control register 2. Don't switch the direction when data is being transferred.

11.3.17 Stop Bit Length

In the UART transmission mode, the stop bit length can be set to either 1 or 2 bits by bit 4
<SBLEN> of the SC0MOD2 register. In the receive side, the stop bit is always regarded as
1 bit.

11.3.18 Status Flag

If the double buffer function is enabled (SC0MOD2 <WBUF> = "1"), the bit 6 flag <RBFLL>
of the SC0MOD2 register indicates the condition of receive buffer full. When one frame of
data has been received and transferred from buffer 1 to buffer 2, this bit is set to "1" to show
that buffer 2 is full (data is stored in buffer 2). When the receive buffer is read by CPU, it is
cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a
status flag. When double buffering is enabled (SC0MOD2 <WBUF> = "1"), the bit 7 flag
<TBEMP> of the SC0MOD2 register indicates that Transmit Buffer 2 is empty. When data is
moved from Transmit Buffer 2 to Transmit Buffer 1 (shift register), this bit is set to "1"
indicating that Transmit Buffer 2 is now empty. When data is set to the transmit buffer by

CPU, the bit is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag.

11.3.19 Configurations of Transmit/Receive Buffer

		<WBUF> = 0	<WBUF> = 1
UART	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O Interface (SCLK input)	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O Interface (SCLK output)	Transmit buffer	Single	Double
	Receive buffer	Single	Double

11.3.20 Software reset

Software reset is generated by writing the bits 1 and 0 of SC0MOD2 <SWRST1:0> as "10" followed by "01". As a result, SC0MOD0<RXE>, SC0MOD1<TXE>, SC0MOD2<TBEMP>,<RBFLL>,<TXRUN> of mode registers and SC0CR<OERR>, <PERR>, <FERR> of control registers and internal circuit is initialized. Other states are maintained.

10.1.1 Interrupt/Error Generation Timing

① RX Interrupts

Fig. 11-10 shows the data flow of receive operation and the route of read.
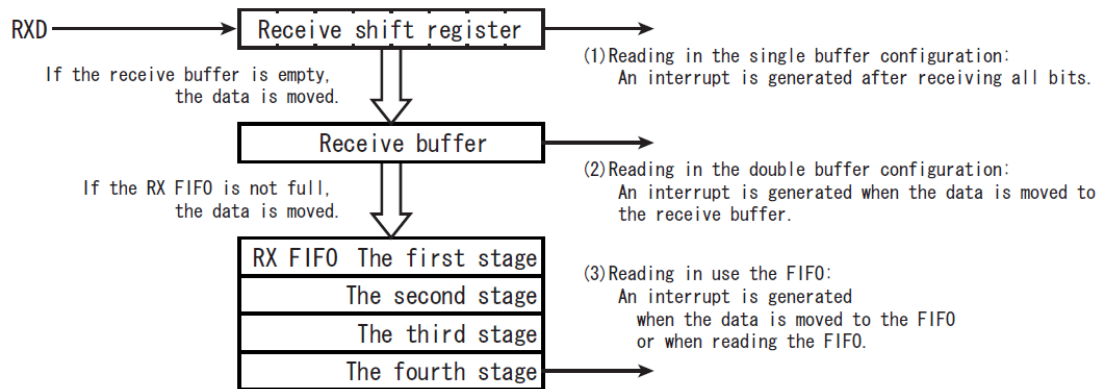

Fig. 11-1 Receive Buffer/FIFO Configuration Diagram

Single Buffer / Double Buffer

RX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configuration	UART mode	IO interface modes
Single Buffer	-	· Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SC0CR<SCLKS> setting.)
Double Buffer	· Around the center of the first stop bit	· Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.) · On data transfer from the shift register to the buffer by reading buffer.

(Note)   Interrupts are not generated when an overrun error is occurred.

FIFO

In use of FIFO, receive interrupt is generated on the condition that the following either operation and SC0RFC<RFIS > setting are established.

● Reception completion of all bits of one frame.

● Reading FIFO

Interrupt conditions are decided by the SCxRFC<RFIS> settings as described in Table 11-5.

Table 11-2 Receive Interrupt conditions in use of FIFO

SC0RFC<RFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is greater than or equal to "the fill level of FIFO intrruption generation."

② TX Interrupts

Fig. 11-11 shows the data flow of transmit operation and the route of write.
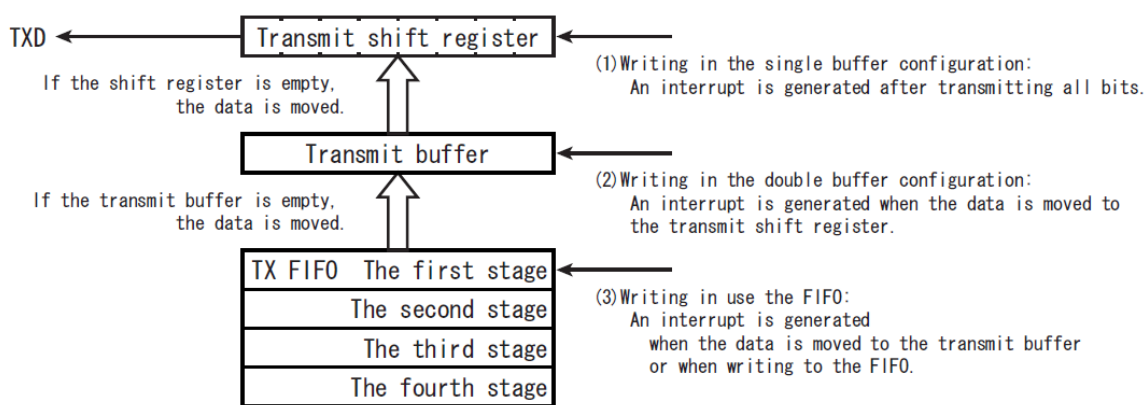
Fig. 11-2 Transmit Buffer/FIFO Configuration Diagram

Single Buffer / Double Buffer

TX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configuration	UART mode	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SC0CR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register.	

(Note)   If double buffer is enabled, a interrupt is also generated when the data is moved from the buffer to the shift register by writing to the buffer.

FIFO

In use of FIFO, transmit interrupt is generated on the condition that the following either operation and SC0TFC<TFIS> setting are established.

● Transmittion completion of all bits of one frame.

● Writing FIFO

Interrupt conditions are decided by the SC0TFC<TFIS> settings as described in Table 11-6.

Table 11-3 Transmit Interrupt conditions in use of FIFO

SC0TFC<TFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is smaller than or equal to "the fill level of FIFO intrruption generation."

③ Error Generation

<u>UART Mode</u>

mode	9 bits	7 bits 8 bits 7bits+Parity 8bits＋Parity
Framing Error Overrun Error	Around the center of stop bit	
Parity Error	-	Around the center of parity bit

<u>I/O Interface Mode</u>

Overrun Error	Immediately after the raising / falling edge of the last SCLK (Rising or falling is determined according to SC0CR<SCLKS> setting.)
Underrun Error	Immediately after the rising or falling edge of the next SCLK. (Rising or falling is determined according to SC0CR<SCLKS> setting.)

11.4 Register Description (Only for Channel 0)

The channel 0 registers are described here. Each register for all the channels operates in the same way.

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

11.4.1 Enable register

		7	6	5	4	3	2	1	0
	bit Symbol								SIOE
SC0EN	Read/Write				R				R/W
	After reset				0				0
	Function	"0" is read.							SIO operation 0:disabled 1:enabled

<SIOE>: Specified the SIO operation.
To use the SIO, enable the SIO operation.
When the operation is disabled, no clock is supplied to the other registers in the SIO module. This can reduce the power consumption.
If the SIO operation is executed and then disabled, the settings of each register except SC0TFC[1:0] are maintained.

(Note)	Do not set SC2EN<SIOE> of SIO2 to "1" in TMPM373/374.

11.4.2 Buffer register

SC0BUF works as a transmit buffer for WR operation and as a receive buffer for RD operation.

		7	6	5	4	3	2	1	0
	bit Symbol	TB7/RB7	TB6/RB6	TB5/RB5	TB4/RB4	TB3/RB3	TB2/RB2	TB1/RB1	TB0/RB0
SC0BUF	Read/Write				R/W				
	After reset	0	0	0	0	0	0	0	0
	Function	TB7 to 0 : Transmit buffer/FIFO RB7 to 0 : Receive buffer/FIFO							

<TB7:0> Transmit buffer (at WR operation).

<RB7:0> Receive buffer (at RD operation).

11.4.3 Control register

SC0CR

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to "0" when read)			R/W	
After reset	0	0	0	0	0	0	0	0
Function	Receive data bit 8 **(For UART)**	Parity **(For UART)** 0: Odd 1: Even	Add parity **(For UART)** 0: Disabled 1: Enabled	0: Normal operation 1: Error Overrun	Parity/ underrun	Framing	0: SCLK0 ⌐‾⌐ 1: SCLK0 ⌐_⌐	**(For I/O interface)** 0:Baud rate generator 1:SCLK0 pin input

<RB8>: 9th bit of the received data in the 9 bits UART mode.

<EVEN>:   Selects even or odd parity.
             "0": odd parity.
             "1": even parity.
             The parity bit may be used only in the 7- or 8-bit UART mode.

<PE>:   Controls enabling/ disabling parity.
             The parity bit may be used only in the 7- or 8-bit UART mode.

<OERR>:   Error flag **(see note)**
<PERR>:   Indicate overrun error, parity error, underrun error and framing error.
<FERR>:

<SCLKS>: The input clock edge used by the data sending and receiving is selected.
             "0": Data transmit/receive at rising edges of SCLK0
             "1": Data transmit/receive at falling edges of SCLK0
             Please set it to "0" at the clock output mode.

<IOC>:   Selects input clock in the I/O interface mode.
             "0": baud rate generator
             "1": SCLK0 pin input.

(Note)   Any error flag is cleared when read.

11.4.4 Mode control register 0

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transmit data bit 8	Handshake function control 0: CTS diable 1: CTS enable	Receive control 0:Reception disabled 1:Reception enabled	Wake-up function 0:Reception disabled 1:Reception enabled	Serial 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		Serial transfer clock **(for UART)** 00: TB4OUT 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock (SCLK0 input)	

SC0MOD0

<TB8>:   Writes the 9th bit of transmit data in the 9 bits UART mode.

<CTSE>:   Controls handshake function.
Setting "1" enables handshake function using $\overline{CTS}$ pin.

<RXE>:   Controls reception **(see note 1).**
Set <RXE> after setting each mode register (SC0MOD0, SC0MOD1 and SC0MOD2).

<WU>:   Controls wake-up function.
This function is available only at 9-bit UART mode.

	9-bit UART mode	Other modes
0	Interrupt when received	don't care
1	Interrupt only when RB9=1	

<SM1:0>:   Specifies transfer mode.

<SC1:0>:   Selects the serial transfer clock in the UART mode.
As for the I/O interface mode, the serial transfer clock can be set in the control register SC0CR. When TMRB output is used for the serial transfer clock, TB4OUT is used in SIO0,1, and TB7OUT is used in SIO2,3.

(Note 1)	**With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2). Then set <RXE> to "1."**
(Note 2)	**Do not clear SC0MOD0<RXE> to "0" during the receiving data.**

11.4.5 Mode control register 1

		7	6	5	4	3	2	1	0
	bit Symbol	I2S0	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	-
SC0MOD1	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	IDLE 0: Stop 1: Start	Transfer mode setting 00: Transfer prohibited 01: Half duplex(RX) 10:Half duplex(TX) 11:Full duplex		Transmit control 0:Disabled 1: Enabled	Interval time of continuous transmission **(for I/O interface)** 000: None 100:8SCLK 001:1SCLK 101:16SCLK 010:2SCLK 110:32SCLK 011:4SCLK 111:64SCLK			Write "0".

<I2S0>: Specifies the IDLE mode operation.

<FDPX1:0>: Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<TXE>: This bit enables transmission and is valid for all the transfer modes (**see note 1**).

<SINT2:0>: Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode. This parameter is valid only for the I/O interface mode when SCLK0 pin input is not selected.

(Note 1) Specify the mode first and then specify the <TXE> bit.
(Note 2) Do not stop the transmit operation (by setting <TXE> = "0") when data is being transmitted.

11.4.6 Mode control register 2

		7	6	5	4	3	2	1	0
	bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
	Read/Write		R				R/W		
SC0MOD2	After reset	1	0	0	0	0	0	0	0
	Function	Transmit buffer empty flag 0: full 1: Empty	Receive Buffer full flag 0: Empty 1: full	In transmission flag 0: Stop 1: Start	STOP bit **(for UART)** 0: 1-bit 1: 2-bit	Setting transfer direction 0: LSB first 1: MSB first	W-buffer 0: Disabled 1: Enabled	SOFT RESET Overwrite "01" on "10"to reset.	

<TBEMP>:   This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1." Writing data again to the double buffers sets this bit to "0."
                 If double buffering is disabled, this flag is insignificant.

<RBFLL>:   This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0."
                 If double buffering is disabled, this flag is insignificant.

<TXRUN>:   This is a status flag to show that data transmission is in progress.
                 <TXRUN> and <TBEMP> bits indicate the following status.

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

<SBLEN>:   This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

<DRCHG>:   Specifies the direction of data transfer in the I/O interface mode. In the UART mode, it is fixed to LSB first.

<WBUF>:   This parameter enables or disables the transmit/receive buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART. When receiving data in the I/O interface mode (I SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.

<SWRST1:0>: Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits and their internal circuits are initialized **(see note 1, 2 and 3)**.

Register	Bit
SC0MOD0	RXE
SC0MOD1	TXE
SC0MOD2	TBEMP,RBFLL,TXRUN,
SC0CR	OERR,PERR,FERR

(Note 1)	**While data transmission is in progress, any software reset operation must be executed twice in succession.**
(Note 2)	**A software reset requires 2 clocks-duration at the time between the end of recognition and the start of execution of software reset instruction.**
(Note 3)	**A software reset initializes other bits. Resetting a mode register and a control register are needed.**

11.4.7 Baud rate generator control register(SC0BRCR)
Baud rate generator control register 2(SC0BRADD)

SC0BRCR

	7	6	5	4	3	2	1	0
bit Symbol	-	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Write "0".	N + (16 − K)/ 16 divider function 0: disabled 1: enabled	Select input clock to the baud rate generator 00: φT1 01: φT4 10: φT16 11: φT64		Division ratio "N" 0000: 16 0001: 1 0010: 2 : 1111: 15			

SC0BRADD

	7	6	5	4	3	2	1	0
bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	"0" is read.				Specify K for the "N + (16 − K)/16" division 0000: Prohibited 0001: K=1 0010: K=2 : 1111: K=15			

<RB0ADDE>: Specifies N + (16-K)/16 division function.
              N + (16-K)/16 division function can only be used in the UART mode.

<RB0CK1:0>: Specifies the baud rate generator input clock.

<RB0S3:0>: Specifies division ratio "N".

<RB0K3:0>: Specifies K for the "N+(16-K)/16" division.

The division ratio of the baud rate generator can be specified in the registers shown above.
Table 11-5 lists the settings of baud rate generator division ratio.

Table 11-7 Setting division ratio

	BR0ADDE=0	BR0ADDE=1 **(Note 1)** (Only UART)
BR0S	Specify "N" **(Note 2) (Note 3)**	
BR0K	No setting required	Setting "K" **(Note 4)**
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 BR0K <BR0ADDE> to "1" after setting the K value to BR0K. The "N + (16 - K)/16" division function can only be used in the UART mode.

(Note 2)   The division ratio 1("0001") and 16("0000") cannot be applied when the "N + (16 - K)/16" division function is used in the UART mode.

(Note 3)   The division ratio 1 ("0001") is available only for using double buffer in the I/O interface mode.

(Note 4)   Specifying "K = 0" is prohibited.

11.4.8 FIFO configuration register

	7	6	5	4	3	2	1	0
bit Symbol	Reserved	Reserved	Reserved	RFST	TFIE	RFIE	RXTXCNT	CNFG
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	**Be sure to write "000".**			Bytes used in RX FIFO 0: Maximum 1:Same as FILL level of RX FIFO	TX interrupt for TX FIFO 0: Disabled 1: Enabled	RX interrupt for RX FIFO 0:Disabled 1: Enabled	Automatic disable of RXE/TXE 0:None 1:Auto disable	FIFO enable 0: Disabled 1: Enabled

(SC0FCNF)

<RFST>:   When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected **(see note)**.
            0: The maximum number of bytes of the FIFO configured (see also <CNFG>).
            1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL1:0>.

<TFIE>:   When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFIE>:   When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<RXTXCNT>: Controls automatic disabling of transmission and reception.
            The mode control register SCOMOD1 <FDPX1:0> is used to set the types of TX/RX.
            Setting "1" enables to operate as follows.

Half duplex RX	When the receive shift register, receive buffer and RX FIFO are filled up to the specified number of valid bytes, SC0MOD0<RXE> is automatically set to "0" to inhibit further reception.
Half duplex TX	When the data transfer in the TX FIFO, the transmit buffer and the transmit shift register is completed, SC0MOD1<TXE> is automatically set to "0" to inhibit further transmission.
Full duplex	When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<CNFG>:   Enables FIFO.
            If enabled, the SCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows:
            (The type of TX/RX can be specified in the mode control register 1
            SC0MOD1<FDPX1:0>).

Half duplex RX	RX FIFO 4byte
Half duplex TX	TX FIFO 4byte
Full duplex	RX FIFO 2byte＋TX FIFO 2byte

(Note1)	**Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.**
(Note2)	**FIFO cannot be used in the 9bit UART mode.**

11.4.9 RX FIFO configuration register

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS					RIL1	RIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0				0	0
Function	RX FIFO clear 1: Clear "0" is read.	Select interrupt generation condition 0: when the data reaches to the specified fill level. 1: when the data reaches to the specified fill level or the data exceeds the specified fill level at the time data is read.	"0" is read.				FIFO fill level to generate RX interrupts 00:4byte (2 Byte at full duplex) 01:1byte 10:2byte 11:3byte	

SC0RFC

<RFCS>:   Clears RX FIFO
          Setting "1" clears RX FIFO and "0" is always read.

<RFIS>:   Specifies the condition of interrupt generation.
          0: An interrupt is generated when it reaches to the specified fill level.
          An interrupt is generated when it is reaches to the specified fill level or if it exceeds the
          specified fill level at the time data is read.

<RIL1:0>: Specifies FIFO fill level**(see note 1).**

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

(Note 1)   RIL1 is ignored when FDPX1:0 = 11 (full duplex).
**(Note 2)   Clearing the receive FIFO should be done after the setting of transfer mode and
             enabling FIFO.**

11.4.10   TX FIFO configuration register

		7	6	5	4	3	2	1	0
	bit Symbol	TFCS	TFIS					TIL1	TIL0
	Read/Write	W	R/W			R			R/W
SC0TFC	After reset	0	0			0		0	0
	Function	TX FIFO clear 1:Clear Always reads "0".	Select interrupt generation condition 0: when the data reaches to the specified fill level. 1: when the data reaches to the specified fill level or the data cannot reach the specified fill level at the time new data is read.	"0" is read.				FIFO fill level to generate TX interrupts. 00:Empty 01:1byte 10:2byte 11:3byte Note: TIL1 is ignored when FDPX1:0=11 (full duplex).	

<TFCS>:   Clears TX FIFO.
          Setting "1" clears TX FIFO and "0" is always read.

<TFIS>:   Selects interrupt generation condition.
          0: An interrupt is generated when the data reaches to the specified fill level.
          1: An interrupt is generated when the data reaches to the specified fill level or the data cannot reach the
          specified fill level at the time new data is read.

<TIL1:0>: Selects FIFO fill level **(see note).**

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

(Note 1)   TIL1 is ignored when FDPX1:0 = 11 (full duplex).
(Note 2)   Clearing the transmit FIFO should be done after the setting of transfer mode and enabling FIFO.
(Note 3)   Please set SC0TFC again when shifting to standby mode (IDLE,STOP) with SC0EN<0>=0 (SIO operation prohibition * clock stop) or SC0MOD1<7>=0 (operation stop * clock stop of the standby mode inside).

11.4.11   RX FIFO status register

	7	6	5	4	3	2	1	0
bit Symbol	ROR					RLVL2	RLVL1	RLVL0
Read/Write	R		R			R		
After reset	0		0			0	0	0
Function	RX FIFO Overrun 1: Generated	"0" is read.				Status of RX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

SC0RST

<ROR>:   Flags for RX FIFO overrun. When the overrun occurs, these bits are set to "1" **(see note).**

<RLVL2:0>:   Shows the fill level of RX FIFO.

(Note)   The <ROR> bit is cleared to "0" when receive data is read from the SC0BUF register.

11.4.12   TX FIFO status register

	7	6	5	4	3	2	1	0
bit Symbol	TUR					TLVL2	TLVL1	TLVL0
Read/Write	R		R			R		
After reset	1		0			0	0	0
Function	TX FIFO Under run 1:Generated Cleared by writing FIFO	"0" is read.				Status of TX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

SC0TST

<TUR>:   Flags for TX FIFO underrun. When the underrun occurs, these bits are set to "1" **(see note).**

<TLVL2:0>:   Shows the fill level of TX FIFO.

(Note)   The <TUR> bit is cleared to "0" when transmit data is written to the SC0BUF register.

11.5 Operation in Each Mode

11.5.1 Mode 0 (I/O interface mode)

Mode 0 consists of two modes, the "SCLK output" mode to output synchronous clock and the "SCLK input" mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

①   Transmitting data

<u>SCLK output mode</u>

In the SCLK output mode, if SC0MOD2<WBUF> is set to "0" and the transmit double buffers are disabled, 8 bits of data are output from the TXD0 pin and the synchronous clock is output from the SCLK0 pin each time the CPU writes data to the transmit buffer. When all data is output, the INTTX0 interrupt is generated.

If SC0MOD2 <WBUF> is set to "1" and the transmit double buffers are enabled, data is moved from Transmit Buffer 2 to Transmit Buffer 1 when the CPU writes data to Transmit Buffer 2 while data transmission is halted or when data transmission from Transmit Buffer 1 (shift register) is completed. When data is moved from Transmit Buffer 2 to Transmit Buffer 1, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to "1," and the INTTX0 interrupt is generated. If Transmit Buffer 2 has no data to be moved to Transmit Buffer 1, the INTTX0 interrupt is not generated and the SCLK0 output stops.

<WBUF>="0" (if double buffering is disabled)

<WBUF>="1" (if double buffering is enabled and there is data in buffer 2)

<WBUF>="1" (if double buffering is enabled and there is no data in buffer 2)
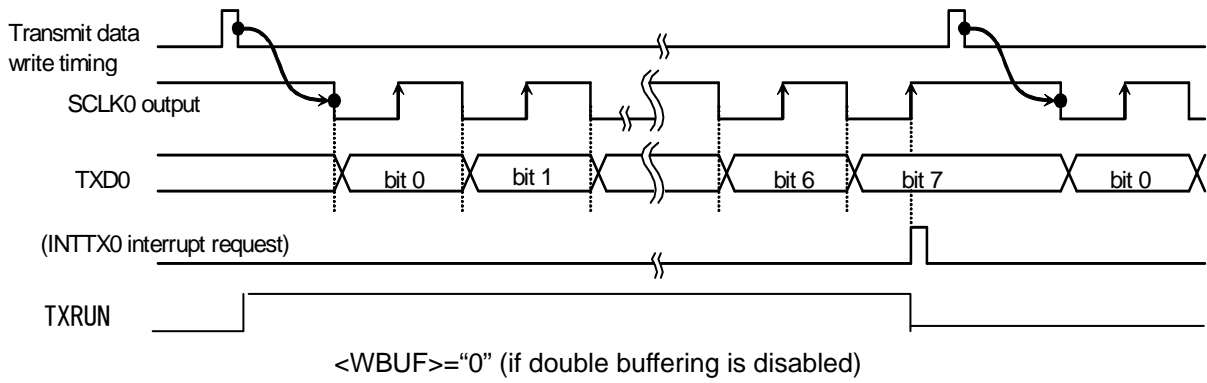
Fig. 11-12 Transmit Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

In the SCLK input mode, if SC0MOD2 <WBUF> is set to "0" and the transmit double buffers are disabled, 8-bit data that has been written in the transmit buffer is output from the TXD0 pin when the SCLK0 input becomes active. When all 8 bits are sent, the INTTX0 interrupt is generated. The next transmit data must be written before the timing point "A" as shown in Fig. 11-13.

If SC0MOD2 <WBUF> is set to "1" and the transmit double buffers are enabled, data is moved from Transmit Buffer 2 to Transmit Buffer 1 when the CPU writes data to Transmit Buffer 2 before the SCLK0 becomes active or when data transmission from Transmit Buffer 1 (shift register) is completed. As data is moved from Transmit Buffer 2 to Transmit Buffer 1, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to "1" and the INTTX0 interrupt is generated. If the SCLK0 input becomes active while no data is in Transmit Buffer 2, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is sent.
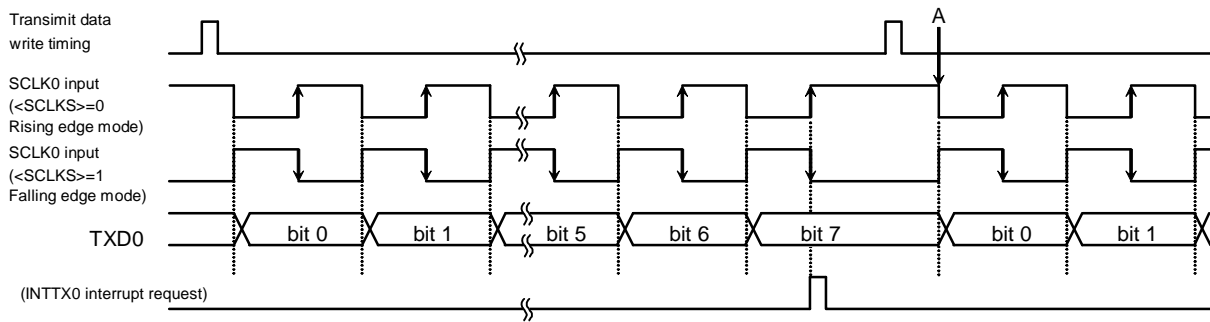
<WBUF>="0" (if double buffering is disabled
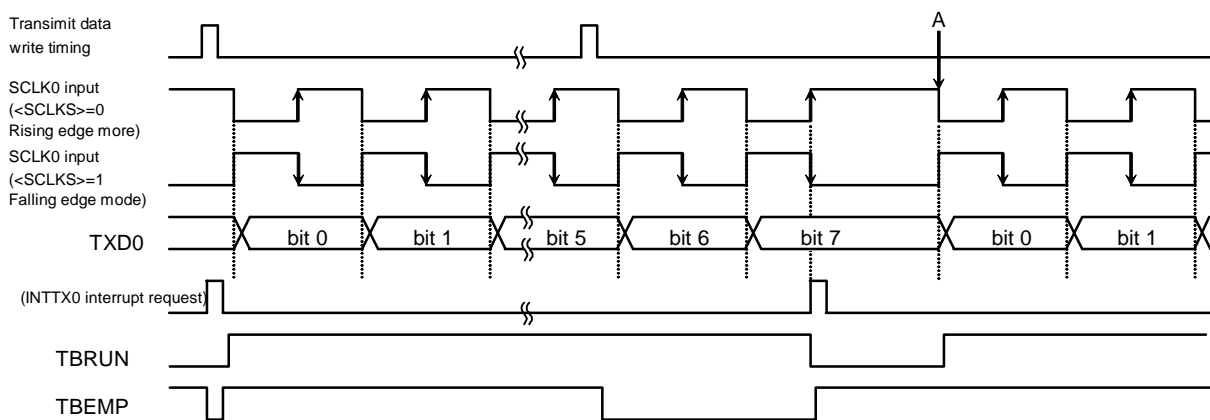
<WBUF>="1" (if double buffering is enabled and there is data in buffer 2)
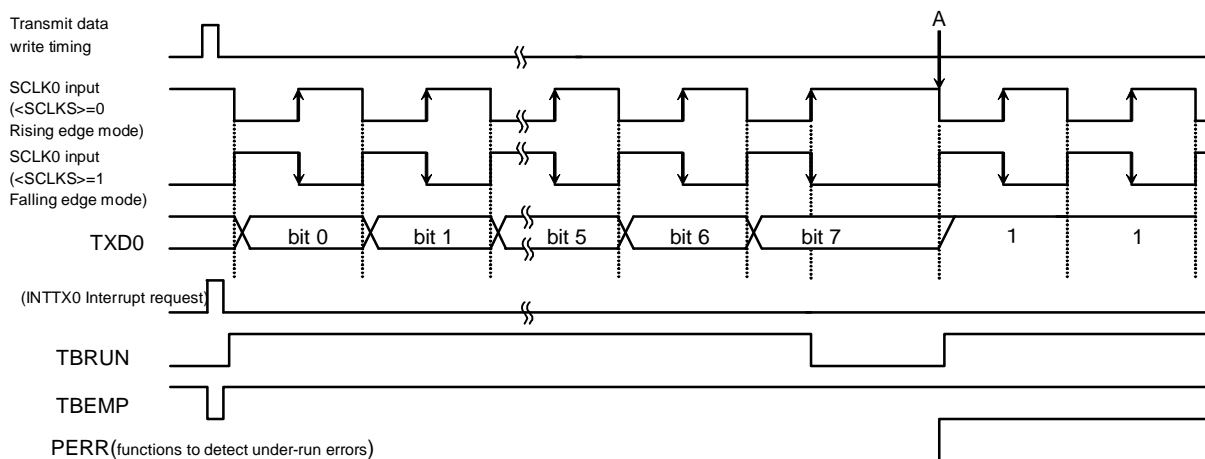
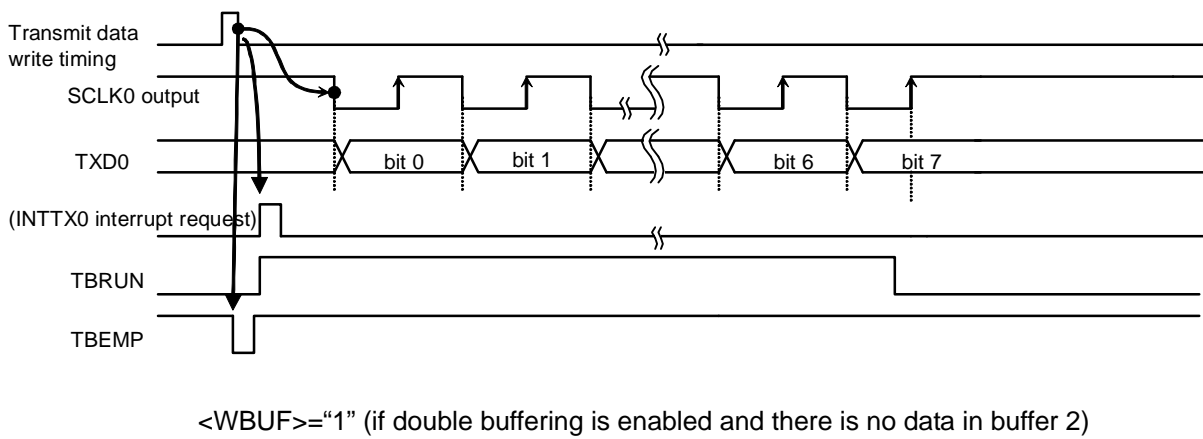<WBUF>="1" (if double buffering is enabled and there is no data in buffer 2)

Fig. 11-13 Transmit Operation in the I/O Interface Mode (SCLK0 Input Mode)

②   Receiving data
SCLK output mode

In the SCLK output mode, if SC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

The first SCLK output can be started by setting the receive enable bit SC0MOD0 <RXE> to "1." If the receive double buffering is enabled with SC0MOD2 <WBUF> set to "1," the first frame received is moved to receive buffer 2 and receive buffer 1 can receive the next frame successively. As data is moved from receive buffer 1 to receive buffer 2, the receive buffer full flag SC0MOD2 <RBFULL> is set to "1" and the INTRX0 interrupt is generated.

While data is in receive buffer 2, if CPU cannot read data from receive buffer 2 before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 2 and thus the INTRX0 interrupt is generated and data reception resumes.
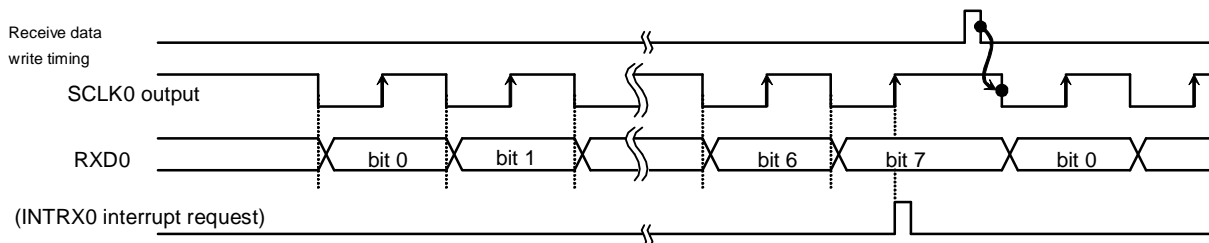
<WBUF>="0" (if double buffering is disabled)
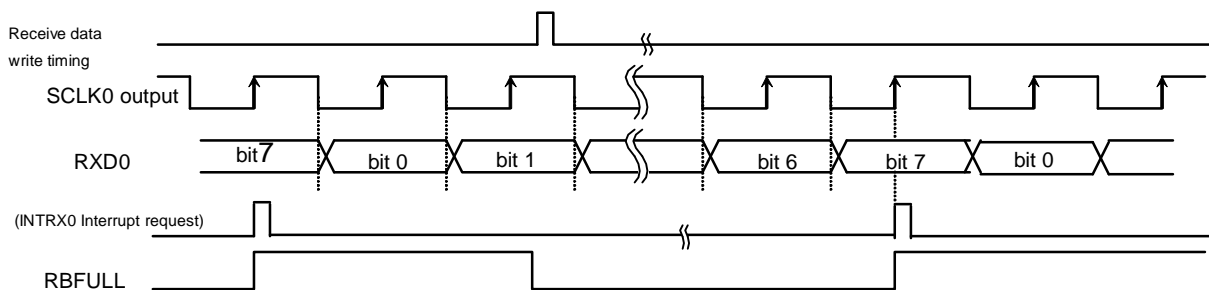
<WBUF>="1" (if double buffering is enabled and data is read from buffer 2)
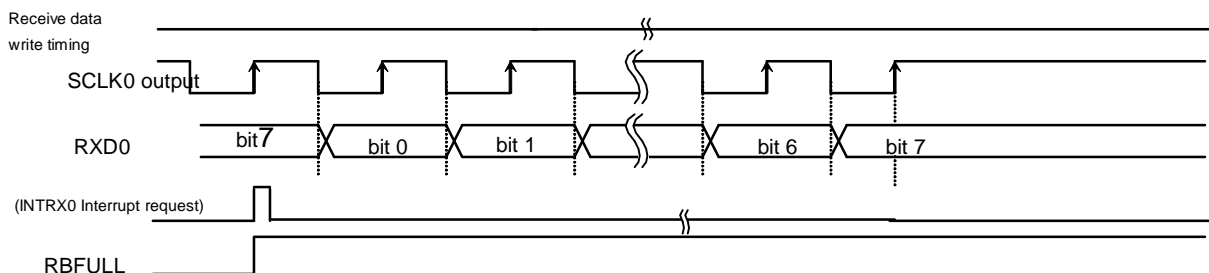
<WBUF>="1" (if double buffering is enabled and data cannot be read from buffer 2)
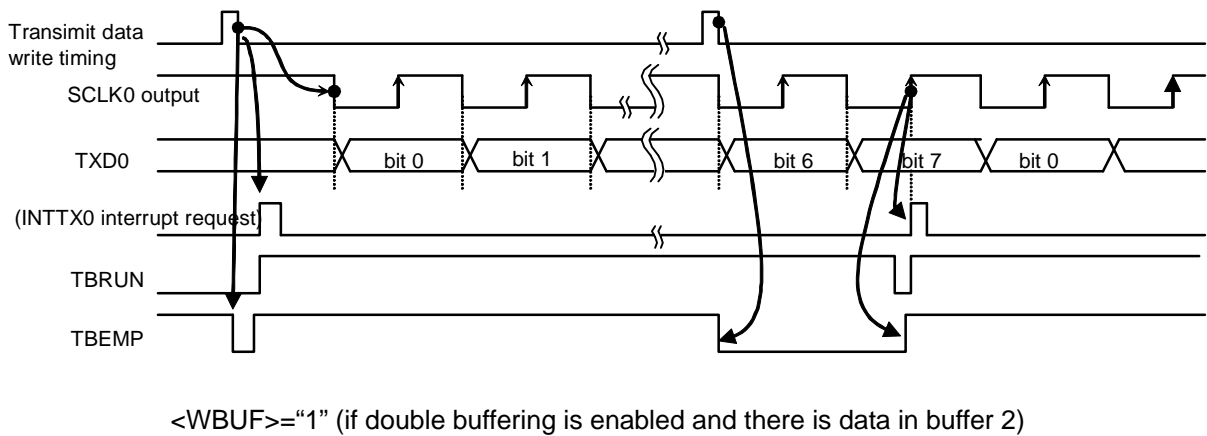
Fig. 11-14 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to receive buffer 2 and receive buffer 1 can receive the next frame successively.

The INTRX receive interrupt is generated each time received data is moved to received buffer 2.
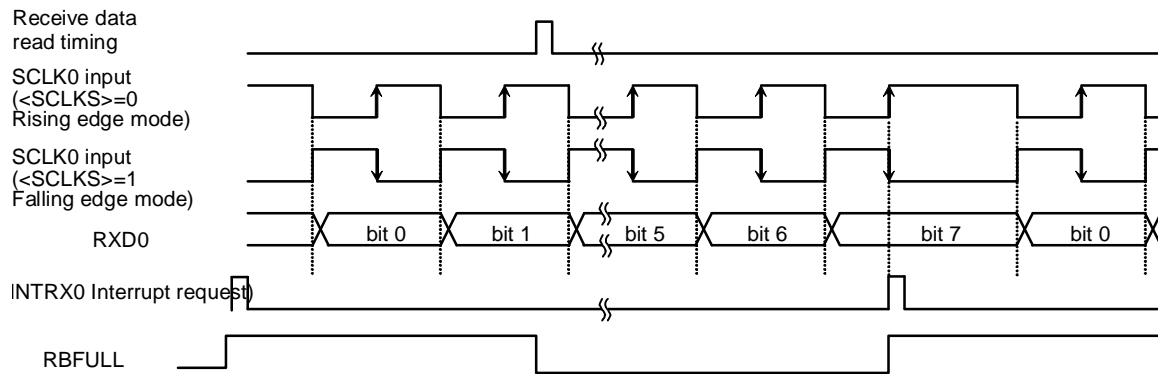
If data is read from buffer 2
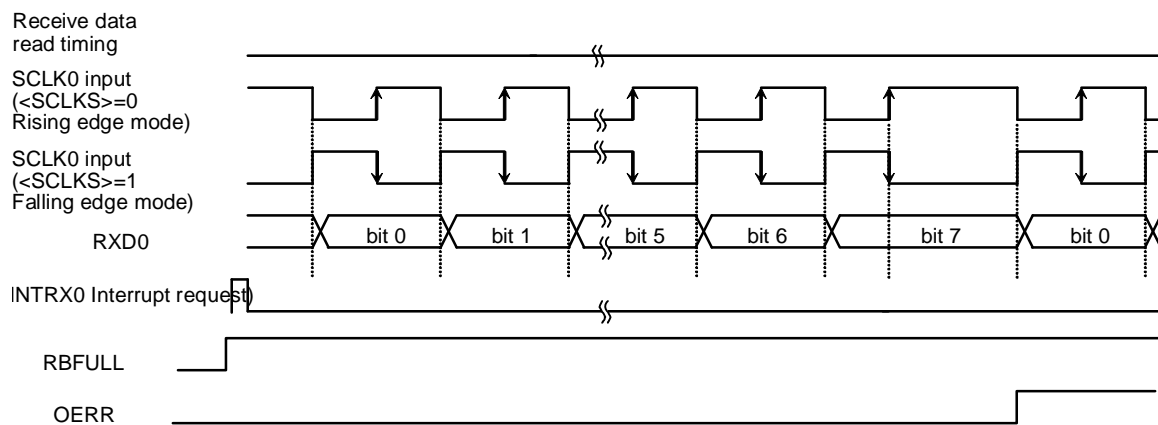
If data cannot be read from buffer 2

Fig. 11-15 Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

(Note) To receive data, SC0MOD <RXE> must always be set to "1" (receive enable) in the SCLK output / SCLK input mode.

③   Transmit and receive (full-duplex)

>   The full-duplex mode is enabled by setting bit 6 <FDPX0> of the serial mode control
>   register 1 (SC0MOD1) to "1".

SCLK output mode

>   In the SCLK output mode, if SC0MOD2 <WBUF> is set to "0" and both the transmit
>   and receive double buffers are disabled, SCLK is output when the CPU writes data to
>   the transmit buffer. Subsequently, 8 bits of data are shifted into receive buffer 1 and the
>   INTRX0 receive interrupt is generated. Concurrently, 8 bits of data written to the
>   transmit buffer are output from the TXD0 pin, the INTTX0 transmit interrupt is
>   generated when transmission of all data bits has been completed. Then, the SCLK
>   output stops. In this, 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 SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and
>   reception, SCLK is output when the CPU writes data to the transmit buffer.
>   Subsequently, 8 bits of data are shifted into receive buffer 1, moved to receive buffer 2,
>   and the INTRX0 interrupt is generated. While 8 bits of data is received, 8 bits of
>   transmit data is output from the TXD0 pin. When all data bits are sent out, the INTTX0
>   interrupt is generated and the next data is moved from the Transmit Buffer 2 to
>   Transmit Buffer 1. If Transmit Buffer 2 has no data to be moved to Transmit Buffer 1
>   (SC0MOD2 <TBEMP> = 1) or when receive buffer 2 is full (SC0MOD2 <RBFULL> = 1),
>   the SCLK clock is stopped. When both conditions, receive data is read and transmit
>   data is written, are satisfied, the SCLK output is resumed and the next round of data
>   transmission is started.

<WBUF>="0" (if double buffering is disabled)

<WBUF>="1" (if double buffering is enabled)

<WBUF>="1" (if double buffering is enabled)
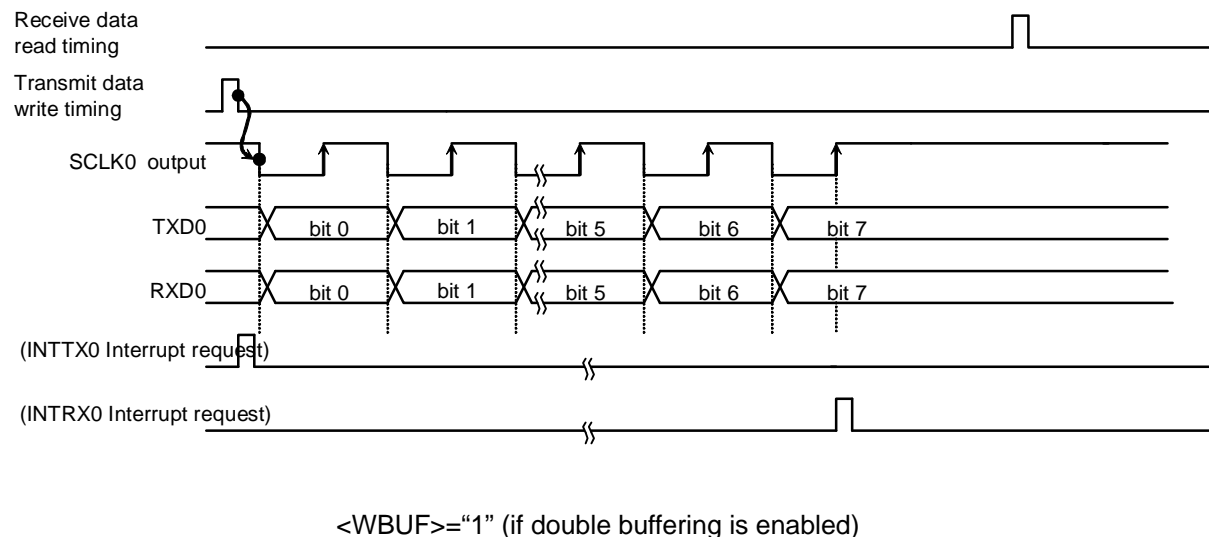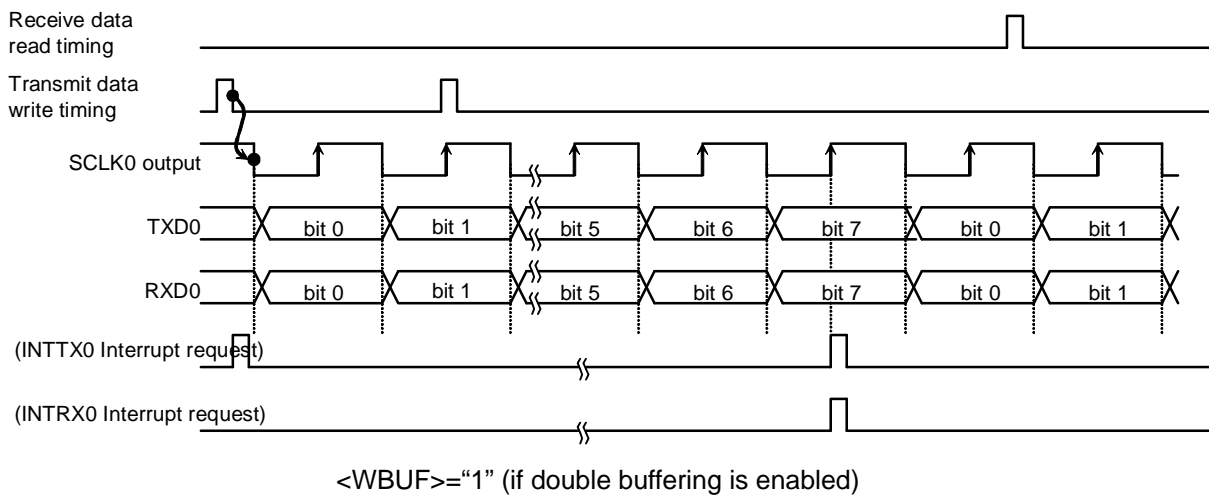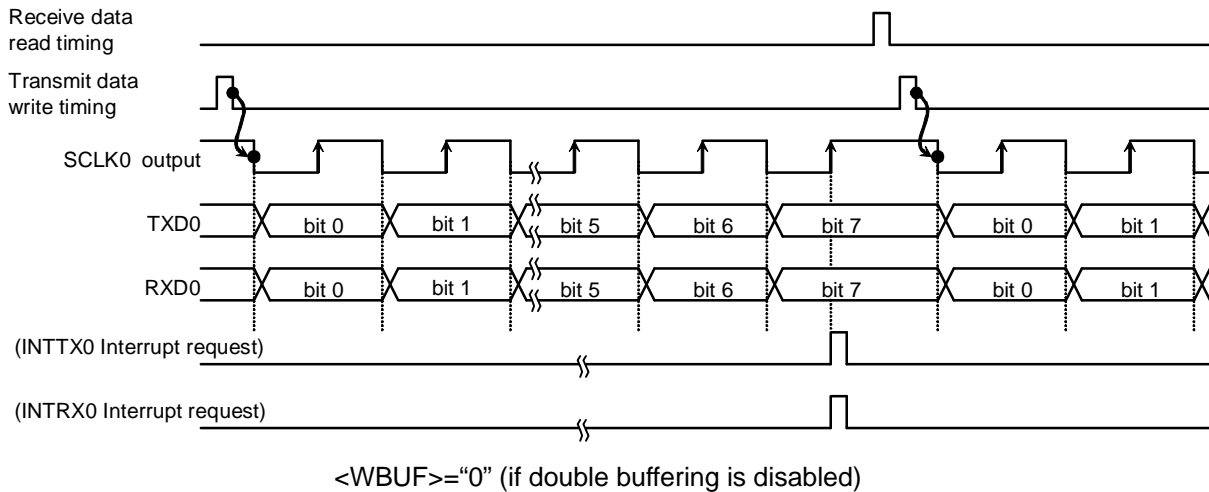
Fig. 11-16 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

In the SCLK input mode with SC0MOD2 <WBUF> set to "0" and the transmit double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the transmit buffer is output from the TXD0 pin and 8 bits of data is shifted into the receive buffer when the SCLK input becomes active. The INTTX0 interrupt is generated upon completion of data transmission and the INTRX0 interrupt is generated at the instant the received data is moved from receive buffer 1 to receive buffer 2. Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Fig. 11-15). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing Transmit Buffer 2 data is moved to Transmit Buffer 1 after completing data transmission from Transmit Buffer 1. At the same time, the 8 bits of data received is shifted to buffer 1, it is moved to receive buffer 2, and the INTRX0 interrupt is generated. Upon the SCLK input for the next frame, transmission from Transmit Buffer 1 (in which data has been moved from Transmit Buffer 2) is started while receive data is shifted into receive buffer 1 simultaneously. If data in receive buffer 2 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 2 when SCLK for the next frame is input, an under-run error occurs.

<WBUF>="0" (if double buffering is disabled)

<WBUF>="1" (if double buffering is enabled with no errors)
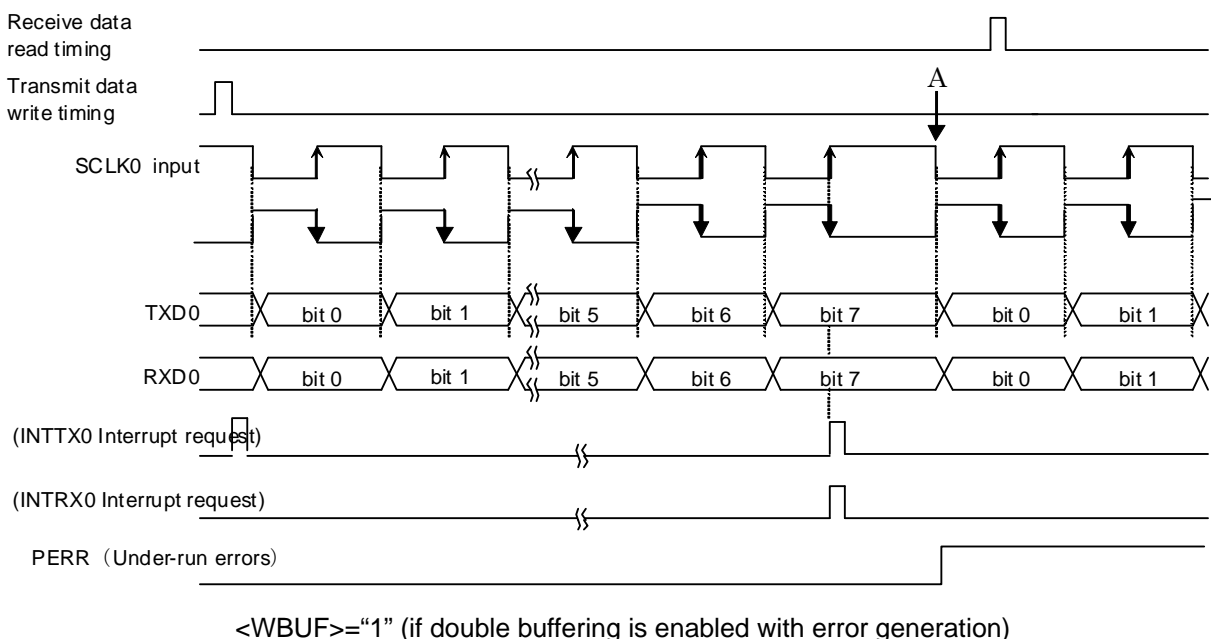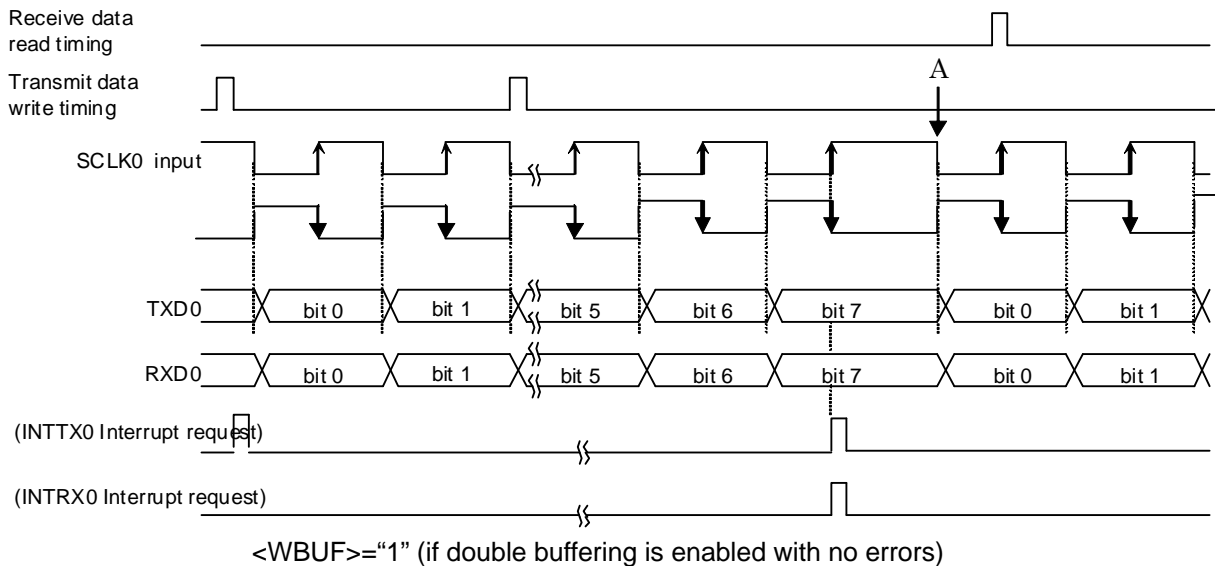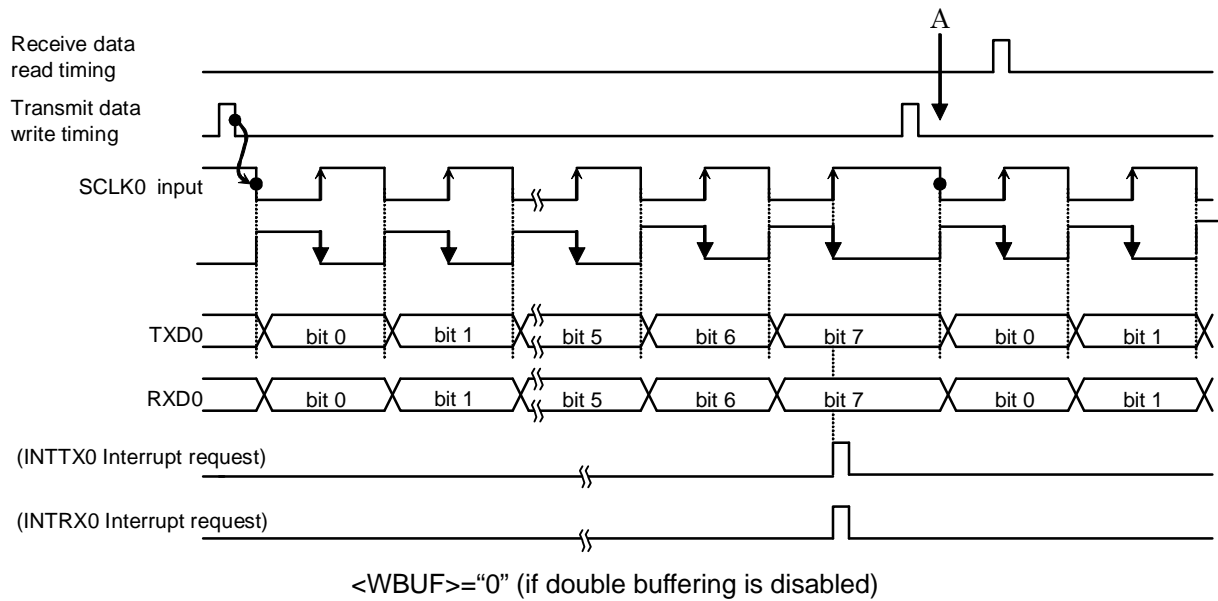
<WBUF>="1" (if double buffering is enabled with error generation)
Fig. 11-17 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

11.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SC0MOD <SM1, 0>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SC0CR <EVEN> bit. The length of the stop bit can be specified using SC0MOD2<SBLEN>.

The following table shows the control register settings for transmitting in the following data format.

Transmission direction (Transmission rate of 2400 bps @ fc = 9.8304 MHz)

* Clocking conditions

System clock	:	high- speed (fc)
High-speed clock gear	:	x1 (fc)
Prescaler clock	:	$f_{periph}/2$ (f_{periph} = fsys)
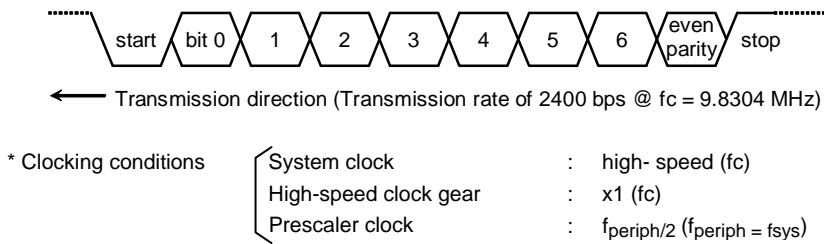
11.5.3 Mode 2 (8-bit UART Mode)

The 8-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SC0CR <PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SC0CR <EVEN>.

The control register settings for receiving data in the following format are as follows:

Transmission direction (transmission rate of 9600 bps @ fc = 9.8304 MHz)

* Clocking conditions

System clock	:	High-speed (fc)
High-speed clock gear	:	x1 (fc)
Prescaler clock	:	$f_{periph}/4$ ($f_{periph} = f_{sys}$)
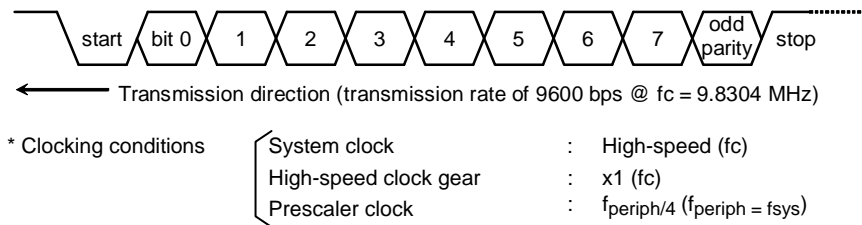
11.5.4 Mode 3 (9-bit UART)

The 9-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "11." In this mode, parity bits must be disabled (SC0CR <PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SC0MOD0) for transmitting data. The data is stored in bit 7 <RB8> of the serial control register SC0CR. 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 SC0BUF. The stop bit length can be specified using SC0MOD2 <SBLEN>.

<u>Wakeup function</u>

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1".
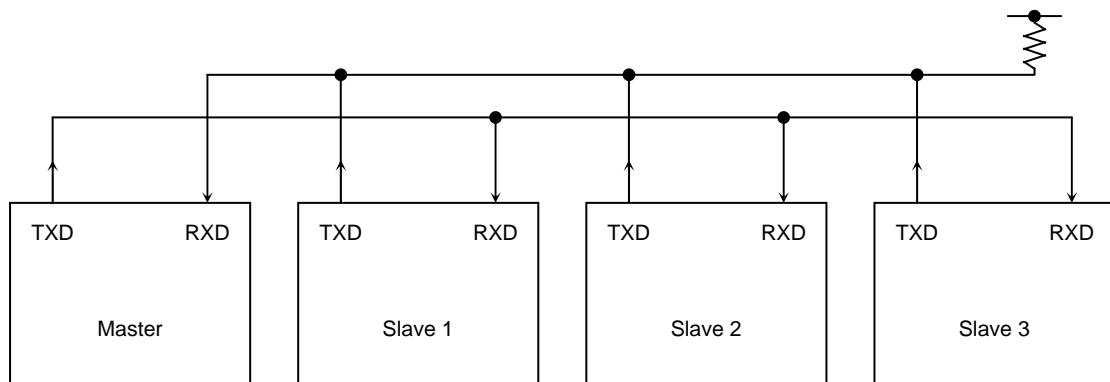
| (Note) | The TXD pin of the slave controller must be set to the open drain output mode using the ODE register. |

Fig. 11-18 Serial Links to Use Wake-up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

② Set SC0MOD <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 SC0MOD <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 (INTRX0) 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.

An example: Using the internal clock f_{SYS} as the transfer clock, two slave controllers are serially linked as follows.
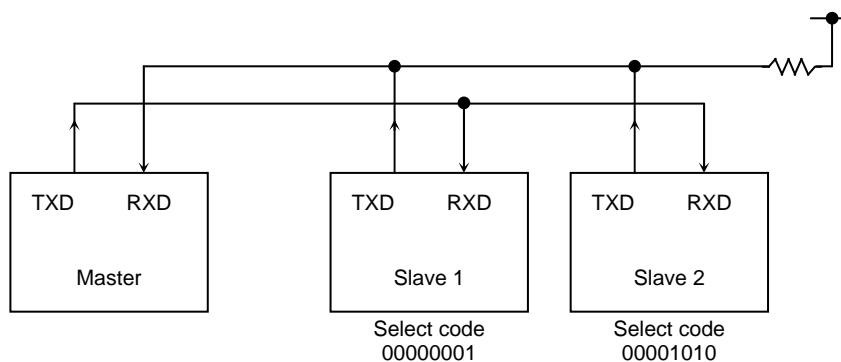
12 Watchdog Timer (Watch Dog Timer)

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 timer detects a runaway, it generates a non-maskable interrupt or an internal reset to notify the CPU. The watchdog timer starts immediately after reset release.

12.1 Configuration

Fig. 12-1 shows the block diagram of the watchdog timer
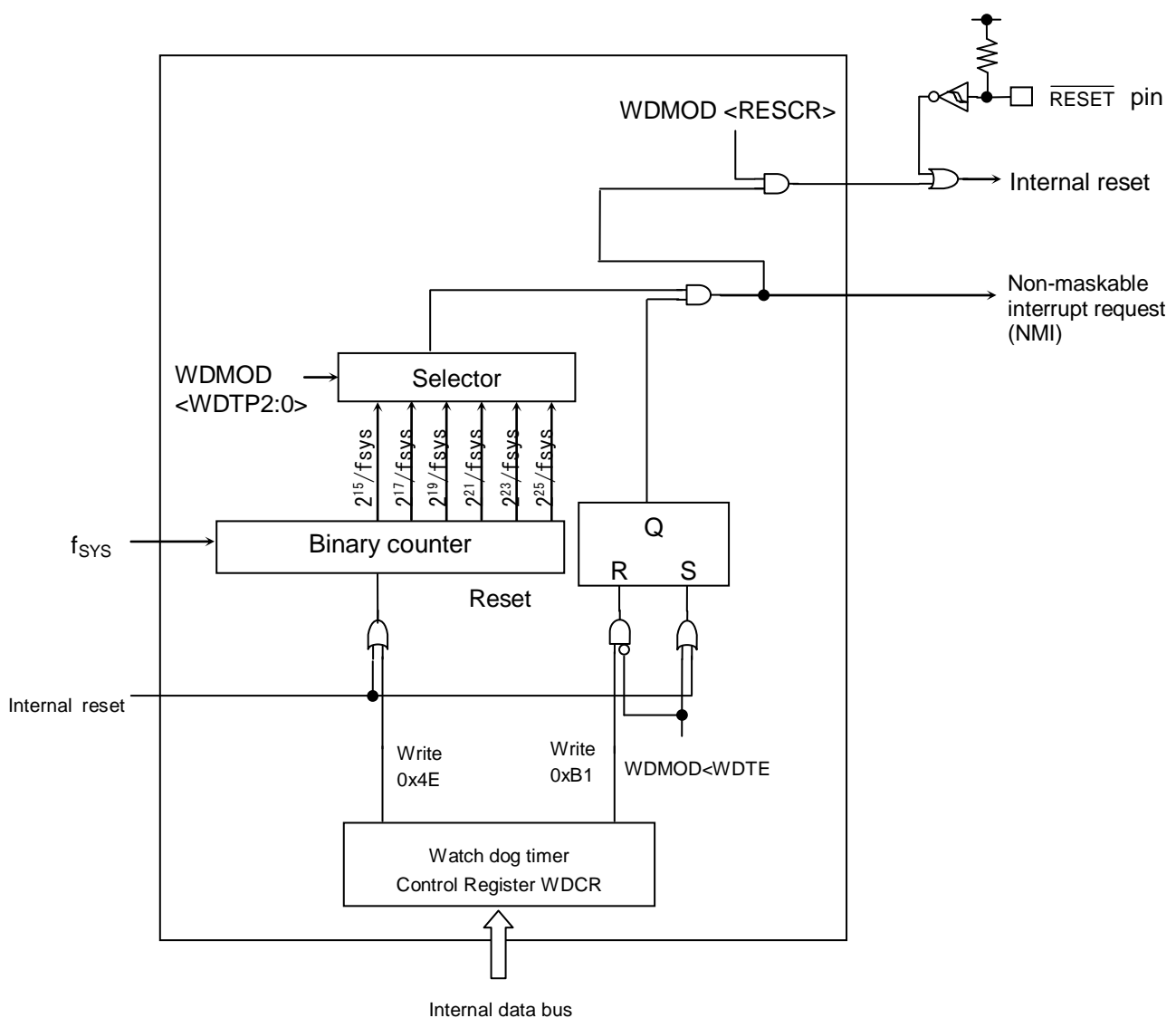
Fig. 12-1 Block Diagram of the Watchdog Timer

12.2 Outline

The watchdog timer consists of the binary counters that are arranged in 25 stages and work using the f_{SYS} system clock as an input clock. The outputs produced by these binary counters are 2^{15}, 2^{17}, 2^{19}, 2^{21}, 2^{23} are 2^{25}.By selecting one of these outputs with WDMOD <WDTP2:0>, NMI can be generated when an overflow occurs, as shown in Fig. 12-2.

12.2.1 Interrupt Mode (WDMOD<RESCR>=0)

When an overflow occurs, the watchdog timer generates NMI to the CPU.
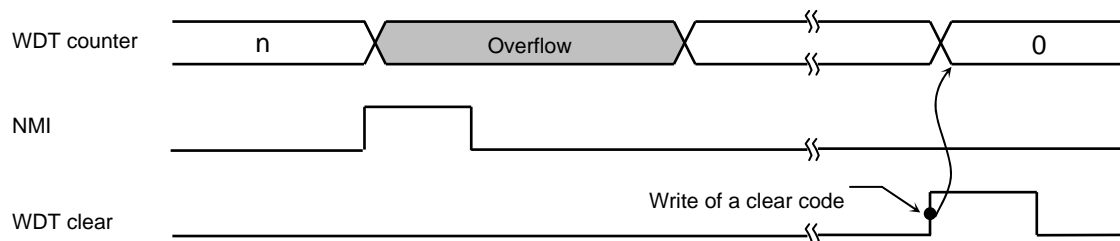
Fig. 12-2 Interrupt Mode

12.2.2 Reset Mode (WDMOD<RESCR>=1)

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is affected for a 32-state time, as shown in Fig. 12-3. If this reset is affected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 1 is used as an input clock f_{SYS}.

32-state
(3.2 μs @ fosc = 10 MHz, f_C = fsys = 10 MHz,)
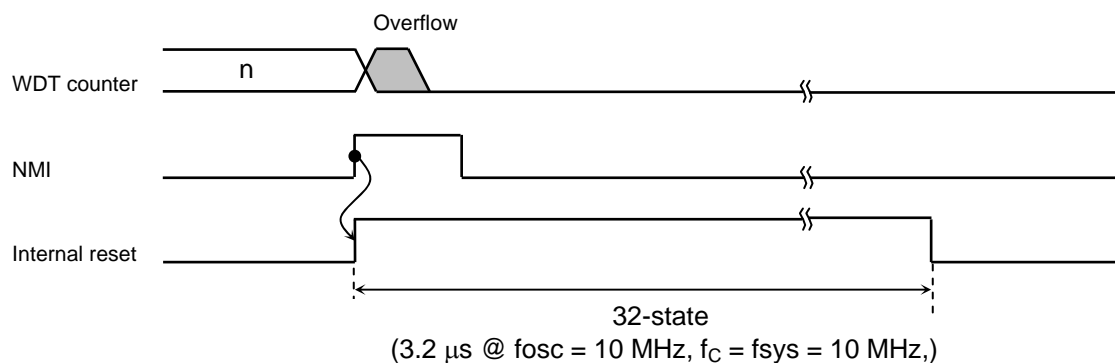
Fig. 12-3 Reset Mode

12.3 Control Registers

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

12.3.1 Watchdog Timer Mode Register (WDMOD)

1. 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, this bit must be set to "0" and, at the same time, the disable code (0xB1) must be written to the WDCR register. This dual setting is intended to minimize the probability that the watchdog timer may inadvertently be disabled if a runaway occurs.

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

2. Specifying the detection time of the watchdog timer <WDTP2: 0>

This is a 3-bit register for specifying the NMI time for runaway detection. When a reset is effected, this register is initialized to WDMOD <WDTP2: 0> = "000." Fig. 12-4 shows the detection time of the watchdog timer.

3. Enabling/disabling the watchdog timer in IDLE mode <I2WDT>

Enabling/disabling the watchdog timer in IDLE mode is controlled by this bit. Writing "1" to this bit enables the watchdog timer and writing "0" to this bit disables the watchdog timer in IDLE mode.

> **(Note)** The watchdog timer is halted in STOP mode.

4. Watchdog timer out reset connection <RESCR>

Setting this bit to "1" enables the watch dog timer to be reset when a runaway is detected. Since a reset initializes this bit to "1," a counter overflow causes a reset.

12.3.2 Watchdog Timer Control Register (WDCR)

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

- Disabling control

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

```
WDMOD   ← 0 – – – – – – –     Clears WDTE to "0."
WDCR    ← 1 0 1 1 0 0 0 1     Writes the disable code (0xB1).
```

- Enabling control

Set WDMOD <WDTE> to "1".

- Watchdog timer clearing control

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

```
WDCR    ← 0 1 0 0 1 1 1 0     Writes the clear code (0x4E)
```

> **(Note)** Writing the disable code (0xB1) clears the binary counter.

Watchdog Timer Mode Register

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP2	WDTP1	WDTP0		I2WDT	RESCR	
Read/Write	R/W	R/W			R	R/W		R/W
After reset	1	0	0	0		0	1	0
Function	WDT control 0: Disable 1: Enable	Selects WDT detection time 000: 2^{15}/f$_{SYS}$ 001: 2^{17}/f$_{SYS}$ 010: 2^{19}/f$_{SYS}$ 011: 2^{21}/f$_{SYS}$ 100: 2^{23}/f$_{SYS}$ 101: 2^{25}/f$_{SYS}$ 110: Setting prohibited 111: Setting prohibited			"0" is read.	IDLE 0: Stop 1: Start	WDT Out control 0: Generates NMI 1: Generates reset	Write "0."

WDMOD (0x4004_0000)

Watchdog timer out control

0	Generates NMI
1	Generates reset

Detection time of watchdog timer   @ fc = 80 MHz

SYSCR1 clock gear value <GEAR2:0>	Detection time of watchdog timer					
	WDMOD<WDTP2:0>					
	000	001	010	011	100	101
000 (fc)	0.41 ms	1.64 ms	6.55 ms	26.21 ms	104.21 ms	419.43 ms
100 (fc/2)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
101 (fc/4)	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
110 (fc/8)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
111 (fc/16)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Watchdog Timer Control Register

	7	6	5	4	3	2	1	0
bit Symbol	-							
Read/Write	W							
After reset	-							
Function	0xB1 : WDT disable code 0x4E : WDT clear code							

WDCR (0x4004_0004)

Disable & clear of WDT

0xB1	WDT disable code
0x4E	WDT clear code
Others	-

Fig. 12-4 Watchdog Timer Registers

12.4 Operation

The watchdog timer generates the NMI or an internal reset after a lapse of the detection time specified by the WDMOD <WDTP2: 0> register. Before generating the NMI or an internal reset, the binary counter for the watchdog timer must be cleared to "0" using software (instruction). If the CPU malfunctions (runaways) due to noise or other disturbances and cannot execute the instruction to clear the binary counter, the binary counter overflows and the non-maskable interrupt by the NMI or an internal reset is generated. The CPU is able to recognize the occurrence of a malfunction (runaway) by identifying the non-maskable interrupt and to restore the faulty condition to normal by using a malfunction (runaway) countermeasure program.

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

In STOP mode, the watchdog timer is reset and in an idle state. In IDLE mode, its operation depends on the WDMOD <I2WDT> setting. Before putting it in IDLE mode, WDMOD <I2WDT> must be set to an appropriate setting, as required.

Example:

1. To clear the binary counter

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

2. To set the detection time of the watchdog timer to $2^{17}/f_{SYS}$.

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

3. To disable the watchdog timer.

   ```
              7 6 5 4 3 2 1 0
   WDMOD  ←   0 − − − − − − −      Clears WDTE to "0"
   WDCR   ←   1 0 1 1 0 0 0 1      Writes the disable code (0xB1)
   ```

(Note 1) The counter of the watchdog timer stops at the debug mode.

13. Oscillation Frequency Detector (OFD)

13.1 Configuration

The oscillation frequency detector generates a reset for I/O if the oscillation of high frequency for CPU clock OFDMNPLLexceeds the detection frequency range.

The oscillation frequency detection is controlled by OFDCR1, OFDCR2 registers and the detection frequency range is specified by OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON which are the detection frequency setting registers. The lower detection frequency is specified by OFDMNPLLOFF/OFDMNPLLON registers and the higher detection frequency is specified by OFDMXPLLOFF/OFDMXPLLON registers.

When the oscillation frequency detection is enabled, writing to OFDMNPLLOFF/OFDMNPLLON/OFDMX-PLLOFF/OFDMXPLLON registers is disabled. Therefore, the setting the detection frequency to these registers should be done when the oscillation frequency detection is disabled. And writing to OFDCR2/OFDMNPLLOFF/ OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers is controlled by OFDCR1 register. To write OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers, the write enable code "0xF9" should be set to OFDCR1 beforehand. To enable the oscillation frequency detector, set "0xE4" to OFDCR2 after setting "0xF9" to OFDCR1. Since the oscillation frequency detection is disabled after an external reset input, power on reset or LVD reset, write "0xF9" to OFDCR1 and write "0xE4" to OFDCR2 register to enable its function.

When the TMPM372/3/4 detects the out of frequency by lower and higher detection frequency setting registers, all I/Os become high impedance by reset. In case of PLLOFF, OFDMNPLLOFF and OFDMXPLLOFF registers are valid for detection and the setting value of OFDMNPLLON/OFDMXPLLON registers are ignored. In case of PLLON, OFDMNPLLON and OFDMXPLLON registers are valid for detection and the setting value of OFDMN-PLLOFF/OFDMXPLLOFF registers are ignored. By the oscillation frequency detection reset, all I/Os except power supply pins, $\overline{\text{RESET}}$, X1 and X2 become high impedance. If oscillation frequency detection reset is generated by detecting the stopping of high frequency, the internal circuities such as registers hold the condition at the timing of oscillation stop. To initialize these internal circuitries, an external re-starting of oscillation is needed.

Since all registers for oscillation frequency detector (OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/ OFDMXPLLOFF/OFDMXPLLON) are not initialized by the reset generated from oscillation frequency detector, the detection of oscillation is keeps its function during the reset period of oscillation frequency detection. Therefore, if the oscillation frequency detection reset occurs, the reset is not released unless the CPU clock resumes its normal frequency.

Note 1: The oscillation frequency detection reset is available only in NORMAL and IDLE modes. In STOP mode, the oscillation frequency detection reset is disabled automatically.

Note 2: When the PLL is controlled (enabled or disabled) by the CGPLLSEL register, the OFD must be disabled beforehand. If OFD reset is generated with PLL-ON, the detection frequency setting registers (OFDMNPLLON/OFDMX-PLLON) are automatically switched over to OFDMNPLLOFF/OFDMXPLLOFF.
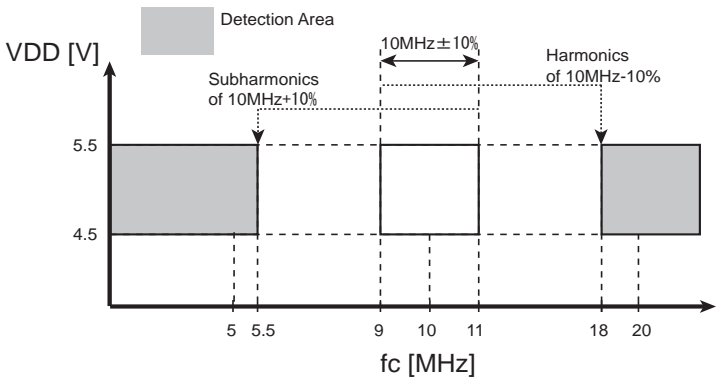
(a) In case of PLLOFF
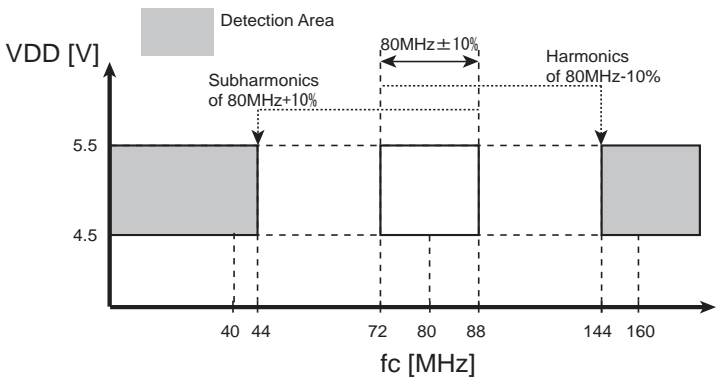
(b) In case of PLLON

Figure 13-1 Example of detection frequency range
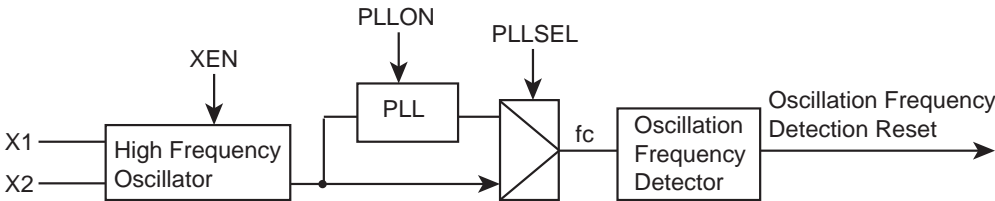
Figure 13-2 Oscillation Frequency Detector

13.2 Control

The oscillation frequency detection is controlled by oscillation frequency detection control register 2 (OFDCR2). The detection frequency is specified by lower/higher detection frequency setting registers (OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON). Writing to OFDCR2/OFDMN-PLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON is controlled by oscillation frequency control register 1 (OFDCR1).

Oscillation frequency detection control register 1

		31-8							
OFDCR1	Bit Symbol	-							
(0x4004_0800)	Read/Write	R							
	After reset	0							
		7	6	5	4	3	2	1	0
	Bit Symbol	CLKWEN7	CLKWEN6	CLKWEN5	CLKWEN4	CLKWEN3	CLKWEN2	CLKWEN1	CLKWEN0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	1	1	0
	Function	0x06: Disabling of writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/ OFDMXPLLON (Write disable code) 0xF9: Enabling of writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/ OFDMXPLLON (Write enable code) Others: Reserved (Note 1)							

Note 1: Only "0x06" and "0xF9" is valid to OFDCR1. If other value than "0x06" and "0xF9" is written to OFDCR1, "0x06" is written to OFDCR1 automatically.

Note 2: OFDCR1 is initialized by the $\overline{RESET}$ pin, power on reset or LVD reset.

Oscillation frequency detection control register 2

		31-8							
OFDCR2	Bit Symbol	-							
(0x4004_0804)	Read/Write	R							
	After reset	0							
		7	6	5	4	3	2	1	0
	Bit Symbol	CLKSEN7	CLKSEN6	CLKSEN5	CLKSEN4	CLKSEN3	CLKSEN2	CLKSEN1	CLKSEN0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	0x00: Disabling of oscillation frequency detection 0xE4: Enabling of oscillation frequency detection Others: Reserved (Note 1)							

Note 1: Only "0x00" and "0xE4" is valid to OFDCR2. Writing other value than "0x00" and "0xE4" to OFDCR2 is ignored.

Note 2: Writing to OFDCR2 is protected by setting "0x06" to OFDCR1 but reading from OFDCR2 is always enabled without setting of OFDCR1.

Note 3: OFDCR2 is initialized by the $\overline{RESET}$ pin, power on reset or LVD reset.

Lower detection frequency setting register (In case of PLL OFF)

		31-9						8	
OFDMNPLLOFF	Bit Symbol	-						OFDMNPLLOFF	
(0x4004_0808)	Read/Write	R						R/W	
	After reset	0						0	
		7	6	5	4	3	2	1	0
	Bit Symbol	OFDMNPLLOFF							
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	1	1	0	1	0

Lower detection frequency setting register (In case of PLL ON)

OFDMNPLLON		31-9						8
	Bit Symbol	-						OFDMNPLLON
(0x4004_080C)	Read/Write	R						R/W
	After reset	0						0
		7	6	5	4	3	2	1	0
	Bit Symbol	OFDMNPLLON							
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	0	1	0	0	0	1

Higher detection frequency setting register (In case of PLL OFF)

OFDMXPLLOFF		31-9						8
	Bit Symbol	-						OFDMXPLLOFF
(0x4004_0810)	Read/Write	R						R/W
	After reset	0						0
		7	6	5	4	3	2	1	0
	Bit Symbol	OFDMXPLLOFF							
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	1	0	1	0	0	0

Higher detection frequency setting register (In case of PLL ON)

OFDMXPLLON		31-9						8
	Bit Symbol	-						OFDMXPLLON
(0x4004_0814)	Read/Write	R						R/W
	After reset	0						1
		7	6	5	4	3	2	1	0
	Bit Symbol	OFDMXPLLON							
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	1	1	1	0	0	1

Note 1: The after reset value is a tentative value.

Note 2: OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON can not be written when the oscillation frequency detection circuit is enabled (OFDCR2="0xE4") or writing is disabled with OFDCR1="0x06". An attempt to write OFDMN-PLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON can not complete a write operation.

Note 3: Writing to OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON is protected by setting "0x06" to OFDCR1 but reading from OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON is always enabled without setting of OFDCR1.

Note 4: Specify an appropriate value to OFDMNPLLOFF and OFDMXPLLOFF depending on the clock frequency to be used under the condition of OFDMNPLLOFF<OFDMXPLLOFF. For how to calculate the value, refer to " 13.3.2 Setting the Lower and Higher Frequency for Detection ".

Note 5: Specify an appropriate value to OFDMNPLLON and OFDMXPLLON depending on the clock frequency to be used under the condition of OFDMNPLLON<OFDMXPLLON. For how to calculate the value, refer to " 13.3.2 Setting the Lower and Higher Frequency for Detection ".

Note 6: OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON are initialized by the $\overline{\text{RESET}}$ pin, power on reset or LVD reset.

Note 7: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

13.3 Function

13.3.1 Enabling and Disabling the Oscillation Frequency Detection

Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection, and writing "0x00" to OFDCR2 with OFDCR1="0xF9" disables the oscillation frequency detection.

Registers of OFD are initialized by the $\overline{\text{RESET}}$ pin, power on reset or LVD reset.

Since OFDCR1 is initialized to "0x06" and OFDCR2 is initialized to "0x00" by resets shown above, oscillation frequency detection and writing to the registers are desabled. Reading from OFDCR2 is always enabled without setting of OFDCR1.

Note: After writing data to OFDCR2, set "0x06" to OFDCR1 to protect OFDCR2 register.

When STOP mode is executed with OFDCR2=0xE4, the oscillation frequency detection is automatically disabled. After releasing STOP and warming up period, the oscillation frequency detection is enabled. The oscillation frequency detection is available only in NORMAL and IDLE mode. Table 13-1 shows the availability of oscillation frequency detector.

Table 13-1 Availability of oscillation frequency detector

Operating Mode	Oscillation Frequency Detection (OFDCR2=0xE4)	All I/Os condition after Oscillation Frequency Detection RESET (Except power supply, $\overline{\text{RESET}}$, X1, X2 pins)
NORMAL	Available	High impedance
IDLE	Available	High impedance
STOP (Including warming up period)	Oscillation Frequency Detection is disabled automatically.	
Reset by oscillation frequency detection reset	Available	High impedance
Watchdog timer reset SYSRESETREQ reset	Available	High impedance
RESET by external reset power on reset LVD reset	Disable	-
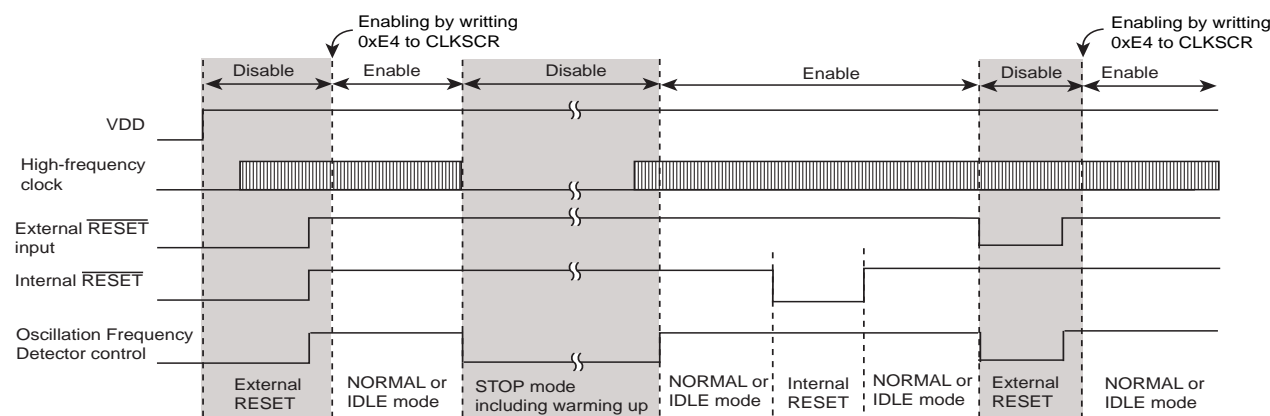
Figure 13-3 Availability of Oscillation Frequency Detection

13.3.2 Setting the Lower and Higher Frequency for Detection

The higher and lower limit of the detection frequency is calculated from the maximum error of the target clock and the reference. The reference clock frequency is 9.5 MHz and the error is $\pm 10\%$.

a)	target clock	Max.
b)		Min.
c)	reference clock	Max. (10.5MHz)
d)		Min. (8.5MHz)

How to calculate the setup value is shown below.

higher limit of the detection frequency $= 1 \div \{ (d \div 2^7) \div (a \div 4) \}$ (truncate after the decimal places)

lower limit of the detection frequency $= 1 \div \{ (c \div 2^7) \div (b \div 4) \}$ (round up after the decimal places)

13.3.3 Oscillation Frequency Detection Reset

If the TMPM372/3/4 detects lower frequency specified by OFDMNPLLOFF/OFDMNPLLON or higher frequency specified by OFDMXPLLOFF/CLKCMXPLLON, the oscillation frequency detector outputs a reset signal for all I/Os.

a. When the high frequency oscillation becomes abnormal

When an abnormal (lower or higher) frequency oscillation continues for some period (T_{OFD}), the oscillation frequency detection reset is generated. By oscillation frequency detection reset initializes all I/Os except power supply pins, $\overline{RESET}$, X1 and X2 become high impedance.

b. When the high frequency oscillation stops

When the high frequency oscillation stops for some period (T_{OFD}), the oscillation frequency detection reset is generated. By oscillation frequency detection reset initializes all I/Os except power supply pins, $\overline{RESET}$, X1 and X2 become high impedance. However, since the internal circuitries such as CPU are initialized by a reset signal latched by high frequency, the internal circuitries hold the state at the oscillation frequency detection.

When the oscillation resumes its normal frequency and continues for some period (T_{OFD}), the oscillation frequency detection reset is released.

· When the high-frequency clock becomes abnormal
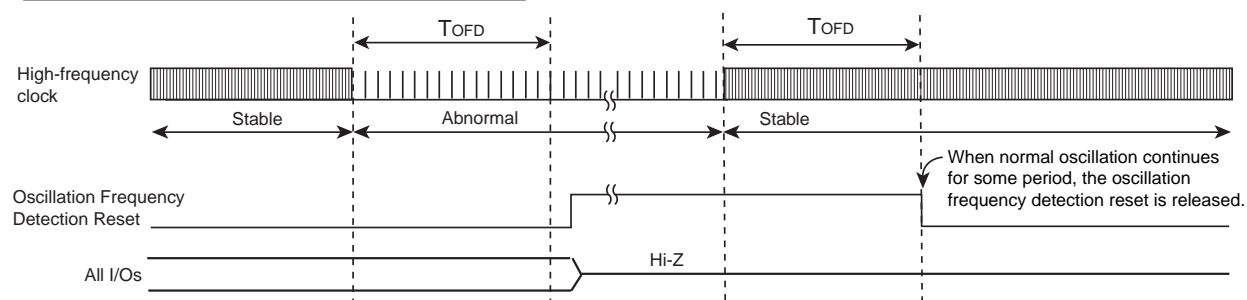
· When the high-frequency clock stops
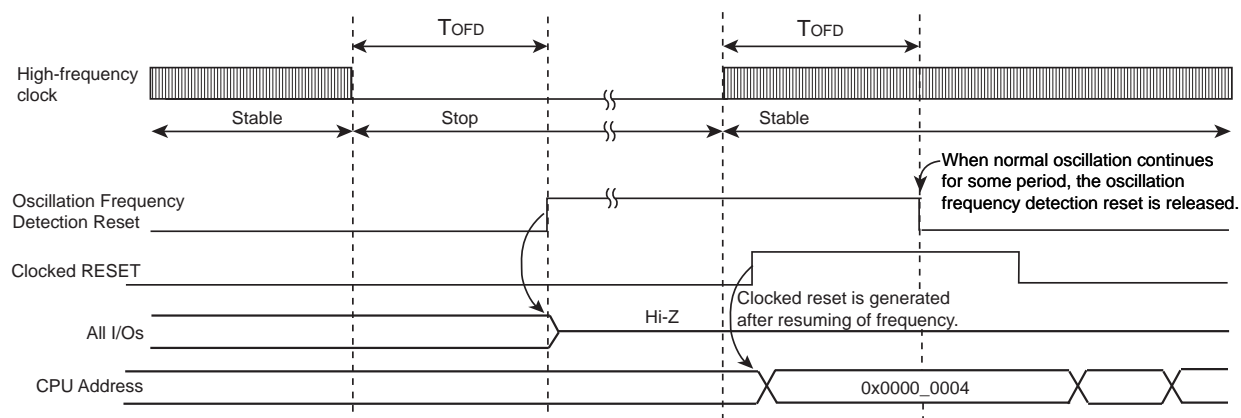
Figure 13-4 Oscillation Frequency Detection Reset Timing

14. Power-on Reset Circuit (POR)

The power-on reset circuit generates a reset when the power is turned on. When the supply voltage is lower than the detection voltage of the power-on reset circuit, a power-on reset signal is generated.

14.1 Configuration

The power-on reset circuit consists of a reference voltage generation circuit, a comparator and a power-on counter.

The supply voltage divided by ladder resistor is compared with the voltage generated by the reference voltage generation circuit by the comparator.
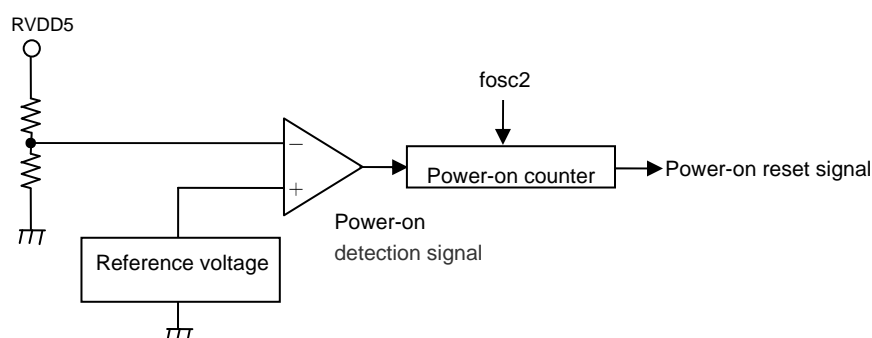
Figure 14-1 Power-on Reset Circuit

14.2 Function

When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated. If the power supply voltage exceeds the releasing voltage of the power-on reset circuit, power-on counter is activated and $2^{15}/f_{osc2}$ (s) later, a power-on reset signal is released.

When power supply voltage goes down, if the supply voltage is equal to or lower than the detecting voltage of the power-on reset circuit, a power-on reset signal is generated.

During the generation of power-on reset, the power-on counter circuit, the CPU and peripheral circuits are reset.

When the power-on reset cuicuit is activated without an external reset input signal, the supply voltage should be increased to the recommended operating voltage range (Note) within 3ms from the detection of the releasing voltage of the power-on reset circuit. If the supply voltage does not reach the range, the TMPM372/3/4 cannot operate properly.
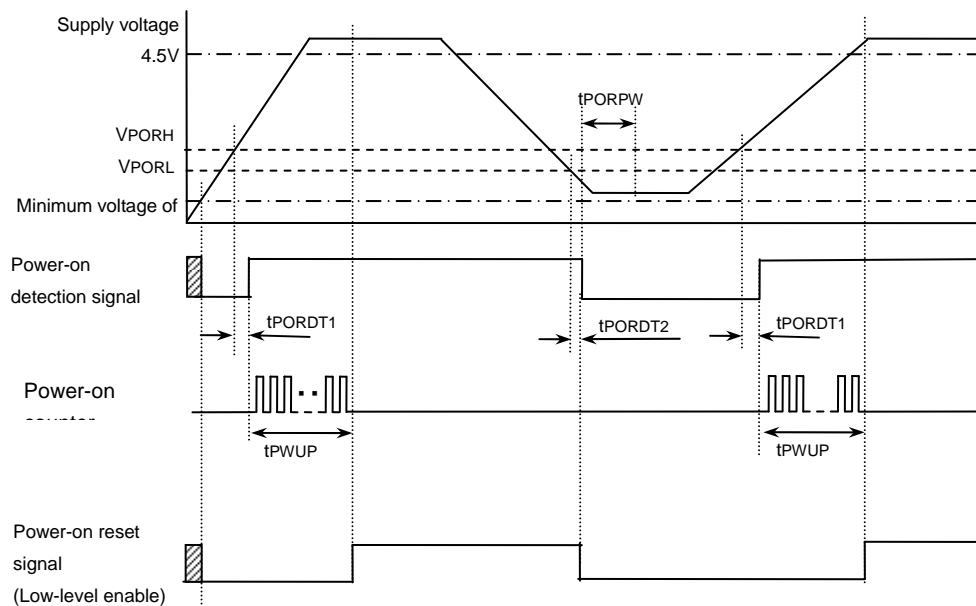
Note 1: The power-on reset circuit may operate improperly, depending on fluctuations in the supply voltage. Refer to the electrical characteristics and take them into consideration when designing equipment.

Note 2: If the supply voltage is lower than the minimum voltage of Power-on Reset circuit in which the circuit cannot operate properly, the power-on reset signal becomes undefined value.

Figure 14-2 Operation Timing of Power-on Reset

Symbol	Parameter	Min	Typ.	Max	Unit
VPORH	Power-on Reset releasing voltage	2.8	3	3.2	V
VPORL	Power-on Reset detection voltage	2.6	2.8	3.0	V
tPORDT1	Power-on Reset release response time		30		µs
tPORDT2	Power-on Reset detection response time		30		µs
tPORPW	Power-on Reset minimum pulse width	45			µs
tPWUP	Power-on counter (Note 2)		$2^{15}/f_{osc2}$		s

Note 1 : Since the power-on reset releasing voltage and the power-on reset detection voltage relatively change, the detection voltage is never reversed.

Note 2 : 3.2ms at 10MHz.

For the details about Power-on sequence, refer to the chapter of "Electrical Characteristics".

For the details about how to use external reset input, refer to "reset exceptions" in the chapter of "Exceptions".

15. Voltage Detection Circuit (VLTD)

The voltage detection circuit detects any decrease in the supply voltage and generates voltage detection reset signals

Note: The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (RVDD5). Refer to the electrical characteristics and take them into consideration when designing equipment.

15.1 Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (RVDD5) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDLVL), and the comparator compares it with the reference voltage.

When the supply voltage (RVDD5) becomes lower than the detection voltage (VDLVL), a voltage detection reset signal is generated.
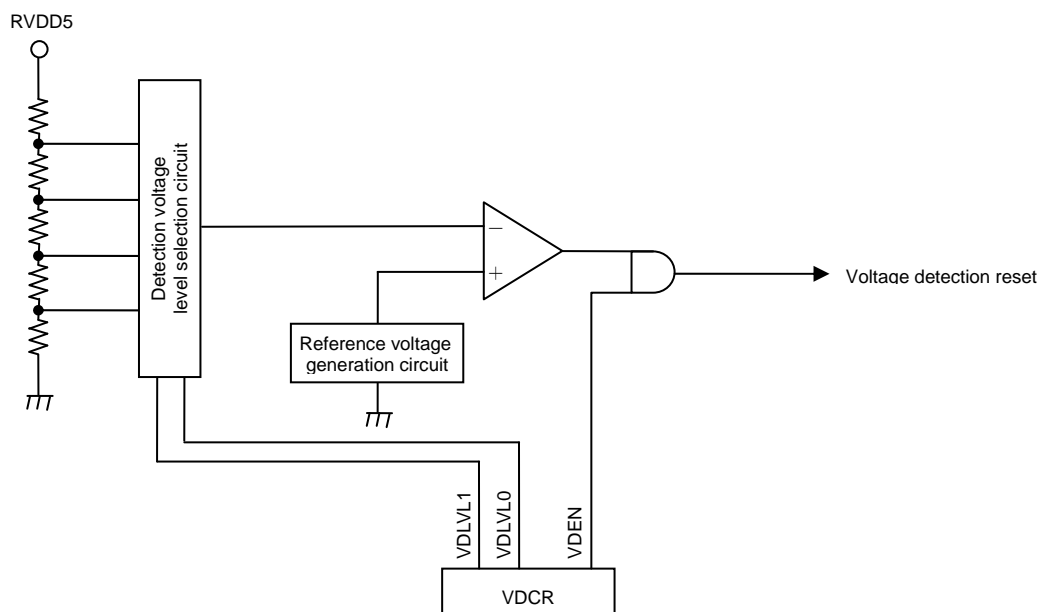
Figure 15-1 Voltage Detection Circuit

15.2 Control

The voltage detection circuit is controlled by voltage detection control registers.

Voltage detection control register

VDCR
(0x4004_0900)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	VDLVL1	VDLVL0	VDEN
Read/Write	R	R	R	R	R	R/W		R/W
After reset	0	0	0	0	0	00		0

VDLVL[1:0]	Selection for detection voltage	00 : Reserved 01 : 4.1 ± 0.2 V 10 : 4.4 ± 0.2 V 11 : 4.6 ± 0.2 V
VDEN	Enables/disables the operation of voltage detection	0 : Disables the operation of voltage detection 1 : Enables the operation of voltage detection

Note 1: VDCR is initialized by a power-on reset or an external reset input.

15.3 Function

The detection voltage can be selected by VDCR<VDLVL[1:0]>. Enabling/disabling the voltage detection can be programmed by VDCR<VDEN>.

After the voltage detection operation is enabled, When the supply voltage (RVDD5) becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection reset signal is generated.

15.3.1 Enabling/disabling the voltage detection operation

Setting VDCR<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

VDCR<VDEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (RVDD5) is lower than the detection voltage (VDLVL), setting VDCR<VDEN> to "1" generates reset signal at the time.

15.3.2 Selecting the detection voltage level

Select a detection voltage at VDCR<VDLVL[1:0]>.
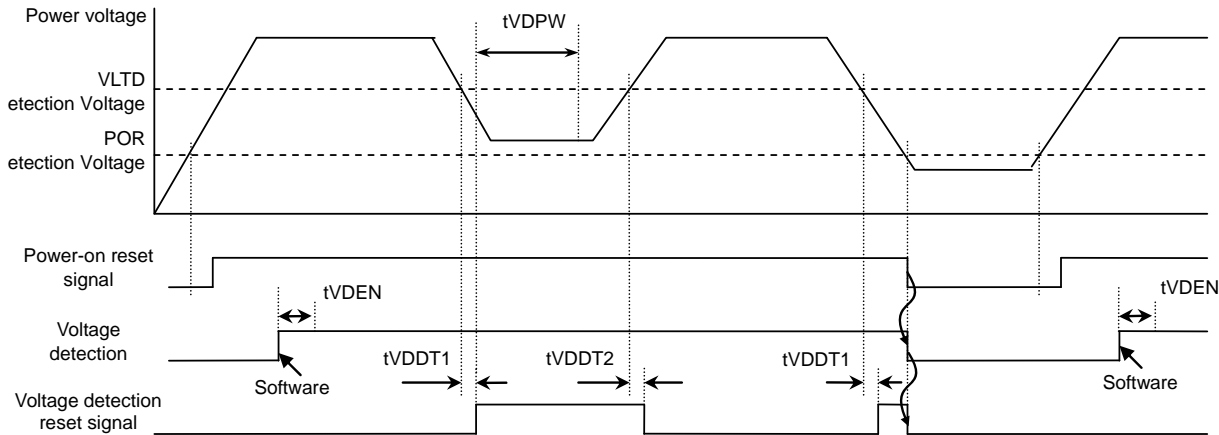
Figure 15-2 Voltage Detection Timing

Symbol	Parameter	Min	Typ.	Max	Unit
tVDEN	Setup time after enabling voltage detection		40		μs
tVDDT1	Voltage detection response time		40		μs
tVDDT2	Voltage detection releasing time		40		μs
tVDPW	Voltage detection minimum pulse width	45			μs

16. Vector Engine (VE)

16.1 Overview

16.1.1 Features

The Vector Engine provides the following features:

1) Executes basic tasks for vector control (coordinate conversion, phase conversion and SIN/COS computation).

   - Uses fixed-point format data.
     → No need for software to manage the decimal point alignment.

2) Enables interface (output control, trigger generation, input processing) with the motor control circuit (PMD: Programmable Motor Driver) and AD converter (ADC).

   - Converts computation results from fixed-point format to data format usable in the PMD.
   - Generates timing data for interactive operation with the PMD and ADC.
   - Converts AD conversion results into fixed-point format.

3) Calculates current, voltage and rotation speed by using normalized values with respect to their maximum values in fixed-point format.

4) Implements PI control in current control.

5) Implements phase interpolation (integration of rotation speed). .
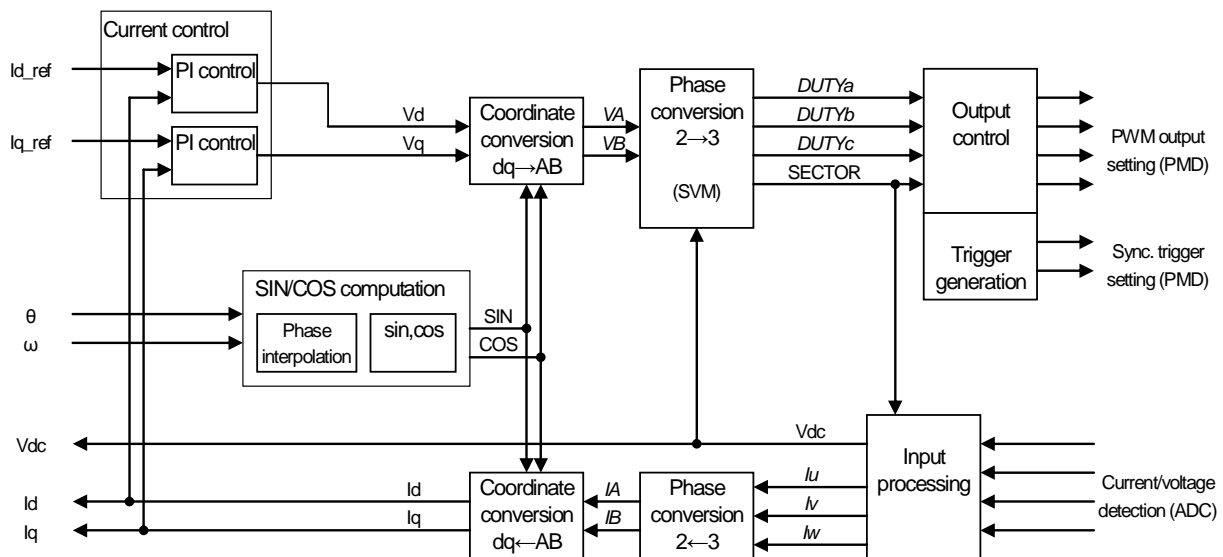
Figure 16-1   Block Diagram of Vector Control

16.1.2 Key Specifications

1) Space vector conversion is used for 2-phase-to-3-phase conversion. Both 2-phase modulation and 3-phase modulation are supported.

2) ADC sampling timing can be generated for sensorless current detection. Current detection can be performed by the 1-shunt, 3-shunt and 2-sensor methods.

3) In current control, PI control is implemented independently for d-axis and q-axis. It is also possible to directly supply reference voltage information without using current control.

4) SIN/COS computations are performed with approximations using series values.
   Phase information can be directly specified or computed from rotation speed by using phase interpolation.

* For using the Vector Engine, the PMD must be set to the VE mode through the mode select register (PMD1MODESEL).
   It is also necessary to make appropriate settings in the ADC (enabling trigger and selecting AIN and result registers to be used) for each synchronizing trigger from the PMD.

16.2 Configuration

Figure 16-2 shows the configuration of the Vector Engine.
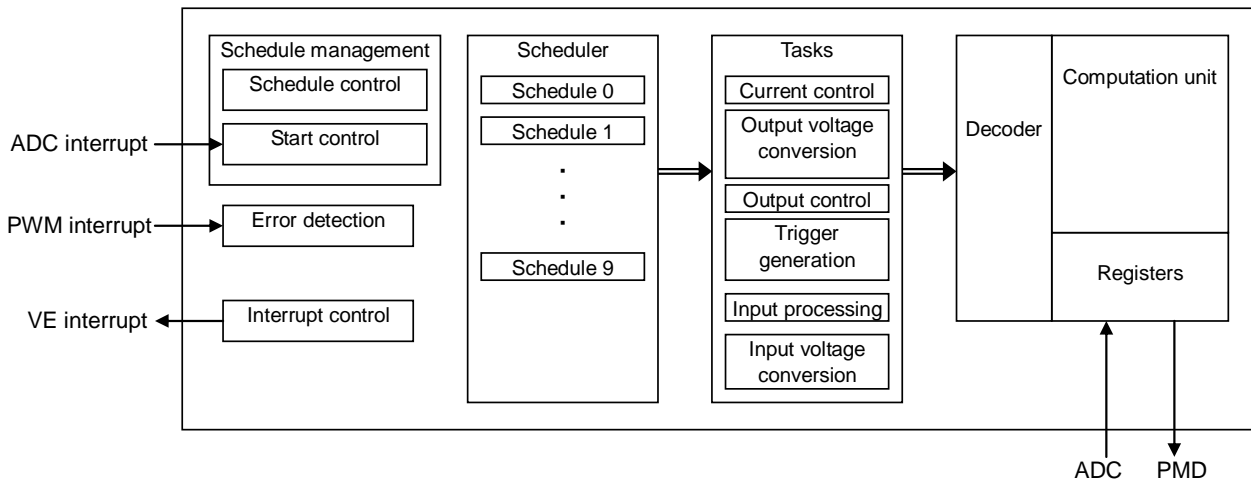
Figure 16-2   Configuration of the Vector Engine

16.2.1   Interaction among Vector Engine, Motor Control Circuit and A/D Converter

As shown in Figure 16-3, the Vector Engine allows direct interaction with the PMD and ADC.

When the PMD1MODESEL register is set to the VE mode, the PMD channel 1 registers PMD1CMPU, PMD1CMPV, PMD1CMPW, PMD1MDOUT, PMD1TRGCMP0, PMD1TRGCMP1 and PMD1TRGSEL are switched to the Vector Engine registers VECMPU1, VECMPV1, VECMPW1, VEOUTCR1, VETRGCMP01, VETRGCMP11 and VETRGSEL1 respectively. In this case, these registers can only be controlled from the Vector Engine, and cannot be written from the PMD. Other PMD registers have no read/write restrictions.

The ADC unit B registers ADBREG0, ADBREG1, ADBREG2, ADBREG3 and UVWISx0, UVWISx1, UVWISx2, UVWISx3 which are the phase information specified by ADBPSETx are read into the Vector Engine as the Vector Engine registers VEADREG0B, VEADREG1B, VEADREG2B, VEADREG3B, VEPHNUM0B, VEPNNUM1B, VEPHNUM2B and VEPHNUM3B respectively. (These registers cannot be accessed from the CPU.) These ADC registers can be written and read from the ADC.

(x=0 to 5)
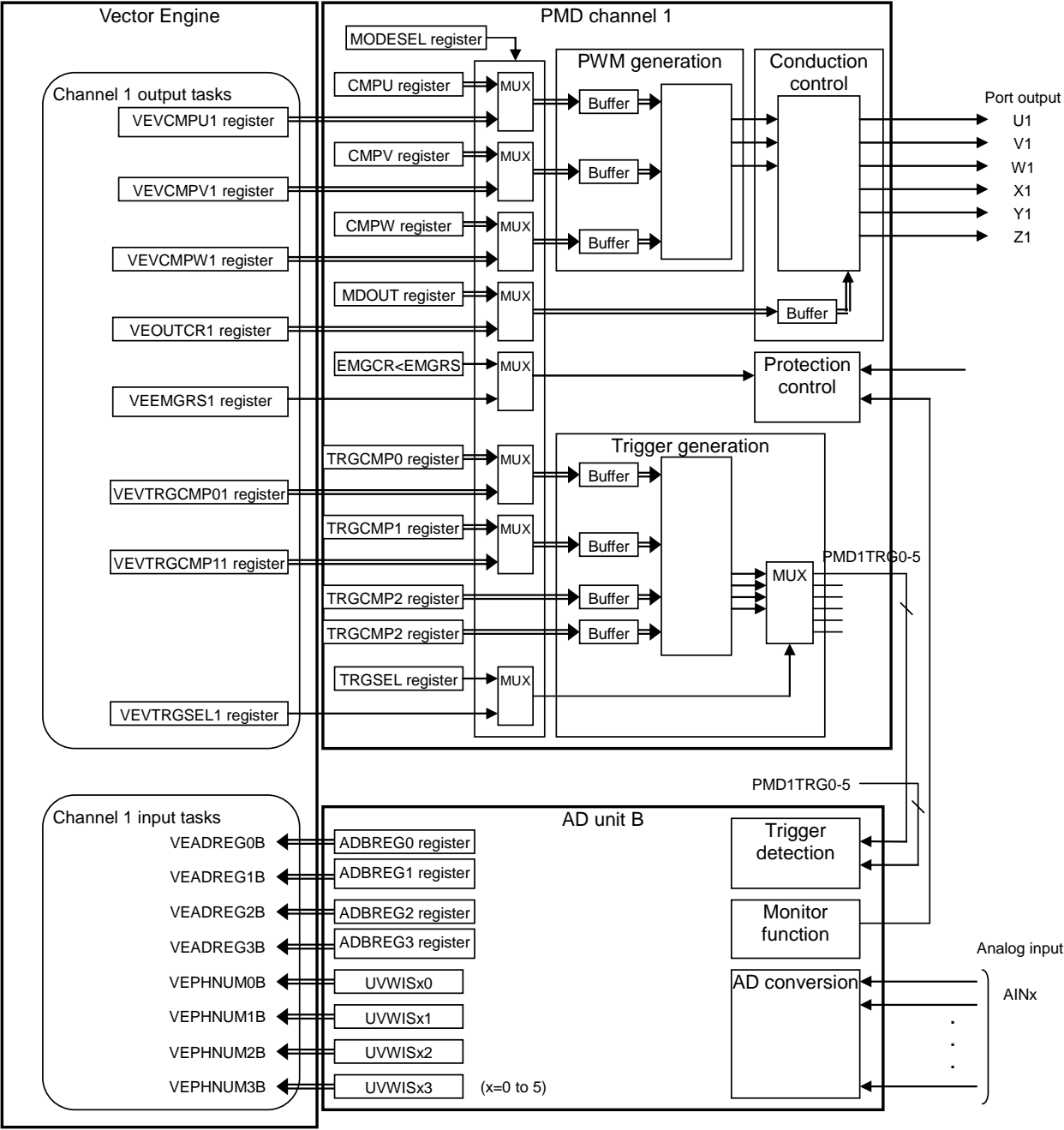
Figure 16-3   Interaction among Vector Engine, PMD and ADC

16.3 List of Registers

The Vector Engine registers are divided into the following three types:

 VE control registers : Vector Engine control registers and temporary registers

 Data and control registers : Computation data and control registers

Table 16-1 List of Registers (1) – VE Control Registers

Address	Register Name	Channel	Description	Read/Write
0x4005-0000	VEEN	-	VE enable/disable	R/W
0x4005-0004	VECPURUNTRG	-	CPU start trigger selection	W
0x4005-0008	VETASKAPP	-	Task selection	R/W
0x4005-000C	VEACTSCH	-	Operation schedule selection	R/W
0x4005-0010	VEREPTIME	-	Schedule repeat count	R/W
0x4005-0014	VETRGMODE	-	Start trigger mode	R/W
0x4005-0018	VEERRINTEN	-	Error interrupt enable/disable	R/W
0x4005-001C	VECOMPEND	-	VE forced termination	W
0x4005-0020	VEERRDET	-	Error detection	R
0x4005-0024	VESCHTASKRUN	-	Schedule executing flag/executing task	R
0x4005-0028	-	-	Reserved	R
0x4005-002C	VETMPREG0	-	Temporary register	R/W
0x4005-0030	VETMPREG1	-	Temporary register	R/W
0x4005-0034	VETMPREG2	-	Temporary register	R/W
0x4005-0038	VETMPREG3	-	Temporary register	R/W
0x4005-003C	VETMPREG4	-	Temporary register	R/W
0x4005-0040	VETMPREG5	-	Temporary register	R/W
0x4005-01BC	-	-	Reserved	R
0x4005-0174	-	-	Reserved	R/W
0x4005-0178	VETADC	Common	ADC conversion time (based on PWM clock)	R/W

Table 16-2   List of Registers (2) – Data and control Registers

Address	Register Name	Description	Read/Write
0x4005-00DC	VEMCTLF1	Status flags	R/W
0x4005-00E0	VEMODE1	Task control mode	R/W
0x4005-00E4	VEFMODE1	Flow control	R/W
0x4005-00E8	VETPWM1	PWM period rate (PWM period [s] × maximum speed[1] × 2^16)	R/W
0x4005-00EC	VEOMEGA1	Rotation speed (speed [Hz]÷ maximum speed [*1] × 2^15)	R/W
0x4005-00F0	VETHETA1	Motor phase (motor phase [deg]/360 × 2^16)	R/W
0x4005-00F4	VEIDREF1	d-axis reference value (current [A] ÷ maximum current[2] × 2^15)	R/W
0x4005-00F8	VEIQREF1	q-axis reference value (current [A] ÷ maximum current [2] × 2^15)	R/W
0x4005-00FC	VEVD1	d-axis voltage (voltage [V] ÷ maximum voltage [3] × 2^31)	R/W
0x4005-0100	VEVQ1	q-axis voltage (voltage [V] ÷ maximum voltage [3] × 2^31)	R/W
0x4005-0104	VECIDKI1	Integral coefficient for PI control of d-axis	R/W
0x4005-0108	VECIDKP1	Proportional coefficient for PI control of d-axis	R/W
0x4005-010C	VECIQKI1	Integral coefficient for PI control of q-axis	R/W
0x4005-0110	VECIQKP1	Proportional coefficient for PI control of q-axis	R/W
0x4005-0114	VEVDIH1	Upper 32 bits of integral term (VDI) of d-axis voltage	R/W
0x4005-0118	VEVDILH1	Lower 32 bits of integral term (VDI) of d-axis voltage	R/W
0x4005-011C	VEVQIH1	Upper 32 bits of integral term (VQI) of q-axis voltage	R/W
0x4005-0120	VEVQILH1	Lower 32 bits of integral term (VQ_I) of q-axis voltage	R/W
0x4005-0124	VEFPWMCHG1	Switching speed (for 2-phase modulation and shift PWM)	R/W
0x4005-0128	VEMDPRD1	PWM period (to be set identically with PMD's PWM period)	R/W
0x4005-012C	VEMINPLS1	Minimum pulse width	R/W
0x4005-0130	VETRGCRC1	Synchronizing trigger correction value	R/W
0x4005-0134	-	Reserved	R/W
0x4005-0138	VECOS1	Cosine value at THETA for output conversion (Q15 data)	R/W
0x4005-013C	VESIN1	Sine value at THETA for output conversion (Q15 data)	R/W
0x4005-0140	VECOSM1	Previous cosine value for input processing (Q15 data)	R/W
0x4005-0144	VESINM1	Previous sine value for input processing (Q15 data)	R/W
0x4005-0148	VESECTOR1	Sector information (0-11)	R/W
0x4005-014C	VESECTORM1	Previous sector information for input processing (0-11)	R/W
0x4005-0150	VEIAO1	AD conversion result of a-phase zero-current [4]	R/W
0x4005-0154	VEIBO1	AD conversion result of b-phase zero-current [4]	R/W
0x4005-0158	VEICO1	AD conversion result of c-phase zero-current [4]	R/W
0x4005-015C	VEIAADC1	AD conversion result of a-phase current [4]	R/W
0x4005-0160	VEIBADC1	AD conversion result of b-phase current [4]	R/W
0x4005-0164	VEICADC1	AD conversion result of c-phase current [4]	R/W
0x4005-0168	VEVDC1	DC supply voltage (voltage [V] ÷ maximum voltage [3] × 2^15)	R/W
0x4005-016C	VEID1	d-axis current (current [A] ÷ maximum current [2] × 2^31)	R/W
0x4005-0170	VEIQ1	q-axis current (current [A] ÷ maximum current [2] × 2^31)	R/W
0x4005-019C	VECMPU1	PMD control: CMPU setting	R/W
0x4005-01A0	VECMPV1	PMD control: CMPV setting	R/W
0x4005-01A4	VECMPW1	PMD control: CMPW setting	R/W
0x4005-01A8	VEOUTCR1	PMD control: Output control (MDOUT)	R/W
0x4005-01AC	VETRGCMP01	PMD control: TRGCMP0 setting	R/W
0x4005-01B0	VETRGCMP11	PMD control: TRGCMP1 setting	R/W
0x4005-01B4	VETRGSEL1	PMD control: Trigger selection	R/W
0x4005-01B8	VEEMGRS1	PMD control: EMG return (EMGCR[EMGRS])	W

*1) Maximum speed: Maximum rotation speed [Hz] that can be controlled or operated

*2) Maximum current: (Phase current value [A] which corresponds to 1 LSB of AD converter) × 2^11

*3) Maximum voltage: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of AD conveter) × 2^12

*4) AD conversion results are stored in the upper 12 bits of each 16-bit register.

16.4 Description of Registers

16.4.1 VE Control Registers

16.4.1.1   VEEN Register

	31		...			9	8
bit Symbol							
Read/Write				R			
After reset				0x00000000			
Function				—			
	7	...	3	2	1		0
bit Symbol					VEIDLEN		VEEN
Read/Write		R			R/W		R/W
After reset		000000			0		0
Function		—			Operation in IDLE mode 0: Inactive 1: Active		VE enable 0: Disable 1: Enable

VEEN (0x4005_0000)

<VEEN>   Disables or enables the Vector Engine.
<VEIDLEN> Controls whether or not the clock is supplied to the Vector Engine in IDLE mode.

16.4.1.2   VECPURUNTRG Register

	31		...			9	8
bit Symbol							
Read/Write				R			
After reset				0x000000			
Function				—			
	7	...	3	2	1		0
bit Symbol					VCPURTB		-
Read/Write		R			W		W
After reset		000000			0		0
Function		—			Vector Engine start command 0: — 1: Start		Write "0"

VECPURUNTRG (0x4005_0004)

<VCPURTB> Starts Vector Engine by programming.

* Always write "0" to bit0.
* When "1" is written to these bits, it is cleared in the next cycle. These bits always read as 0.
* The task to be performed is determined by the settings of the ACTSCH and TASKAPP registers.
* If a channel under executing will be restarted, it must be terminated by COMPEND register before a start command executed.

16.4.1.3   VETASKAPP Register

VETASKAPP (0x4005_0008)		31			・・・			9	8
	bit Symbol								
	Read/Write				R				
	After reset				0x000000				
	Function				—				
		7	6	5	4	3	2	1	0
	bit Symbol		VTASKB				-		
	Read/Write		R/W				R/W		
	After reset		0x0				0x0		
	Function	Task selection 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 -0xF: Reserved				Write "0x0".			

<VTASKB> Specifies the task to be performed when the vevtor engine is started by programming.

   * Only those tasks that are included in schedules can be specified.

16.4.1.4   VEACTSCH Register

VEACTSCH (0x4005_000C)		31			・・・			9	8
	bit Symbol								
	Read/Write				R				
	After reset				0x000000				
	Function				—				
		7	6	5	4	3	2	1	0
	bit Symbol		VACTB				-		
	Read/Write		R/W				R/W		
	After reset		0x0				0x0		
	Function	Vector Engine schedule 0x0: Individual task execution 0x1: Schedule 1 0x4: Schedule 4 0x9: Schedule 9 Other: Reserved				Write "0x0".			

<VACTB> Specifies an individual task execution or a schedule for Vector Engine.

16.4.1.5 VEREPTIME Register

		31				. . .		9	8
VEREPTIME	bit Symbol								
(0x4005_0010)	Read/Write					R			
	After reset					0x000000			
	Function					—			
		7	6	5	4	3	2	1	0
	bit Symbol		VREPB				-		
	Read/Write		R/W				R/W		
	After reset		0x0				0x0		
	Function	Vector Engine repeat count 0: Do not execute schedule 1-15: Execute schedule a specified number of times				Write "0x0".			

<VREPB> Specifies the repeat times a schedule is to be executed in Vector Engine.
  * When "0" is set, no schedule is executed.

16.4.1.6 VETRGMODE Register

		31				. . .		9	8
VETRGMODE	bit Symbol								
(0x4005_0014)	Read/Write					R			
	After reset					0x000000			
	Function					—			
		7	6	5	4	3	2	1	0
	bit Symbol						VTRGB		-
	Read/Write		R				R/W		R/W
	After reset		0x0				00		00
	Function		—			Vector Engine trigger mode Write "10".		Write "00".	

<VTRGB> Specifies the AD conversion end interrupt that triggers input processing in Vector Engine.
  When the VE is used, always write "10" to VTRGB.

16.4.1.7　VEERRINTEN Register

	31	· · ·			9	8
bit Symbol						
Read/Write	R					
After reset	0x000000					
Function	—					

	7	· · ·	3	2	1	0
bit Symbol					VERRENB	-
Read/Write	R				R/W	R/W
After reset	000000				0	0
Function	—				Vector Engine error interrupt enable 0: Disable 1: Enable	Write "0".

VEERRINTEN
(0x4005_0018)

<VERRENB> Enables or disables the error detection interrupt in Vector Engine.

16.4.1.8　VECOMPEND Register

	31	· · ·			9	8
bit Symbol						
Read/Write	R					
After reset	0x000000					
Function	—					

	7	· · ·	3	2	1	0
bit Symbol					VCENDB	-
Read/Write	R				W	W
After reset	000000				0	0
Function	—				Vector Engine forced termination 0: — 1: Terminate	Write "0".

VECOMPEND
(0x4005_001C)

<VCENDB> Forcefully terminates the currently executing schedule in Vector Engine.
*　When "1" is written to these bits, it is cleared in the next cycle. These bits always read as "0".

16.4.1.9   VEERRDET Register

	31	· · ·		9	8
VEERRDET (0x4005_0020) bit Symbol					
Read/Write			R		
After reset			0x000000		
Function			—		

	7	· · ·	3	2	1	0
bit Symbol					VERRDB	-
Read/Write		R			R	R
After reset		000000			0	0
Function		—			Channel 1 error flag 0: No error detected 1: Error detected	

<VERRDB> Vector Engine error flag
The error flags are set when a PWM interrupt is detected during execution of a schedule (excluding standby periods waiting for a start trigger).

* The error flags are cleared by a read of this register.

16.4.1.10   VESCHTASKRUN Register

	31	· · ·		17	16
VESCHTASKRUN (0x4005_0024) bit Symbol					
Read/Write			R		
After reset			0x0000		
Function			—		

	15	· · ·		11	10
bit Symbol					
Read/Write			R		
After reset			000000		
Function			—		

	9	8	7	6	5	4	3	2	1	0
bit Symbol		VRTASKB			VRSCHB			-		-
Read/Write		R			R		R			R
After reset		0x0			0		0x0			0
Function	Task executing in channel 1 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 -0xF: Reserved				Channel 1 schedule status 0: Not executing 1: Executing					

<VRSCHB> Schedule execution status in Vector Engine
<VRTASKB> Task Number currently executing in Vector Engine

16.4.1.11 Temporary Registers

VETMPREG0		31	· · ·	1	0
(0x4005_002C)	bit Symbol		TMPREG0		
	Read/Write		R/W		
	After reset		0x00000000		
	Function				

VETMPREG1		31	· · ·	1	0
(0x4005_0030)	bit Symbol		TMPREG1		
	Read/Write		R/W		
	After reset		0x00000000		
	Function				

VETMPREG2		31	· · ·	1	0
(0x4005_0034)	bit Symbol		TMPREG2		
	Read/Write		R/W		
	After reset		0x00000000		
	Function				

VETMPREG3		31	· · ·	1	0
(0x4005_0038)	bit Symbol		TMPREG3		
	Read/Write		R/W		
	After reset		0x00000000		
	Function				

VETMPREG4		31	· · ·	1	0
(0x4005_003C)	bit Symbol		TMPREG4		
	Read/Write		R/W		
	After reset		0x00000000		
	Function				

VETMPREG5		31	· · ·	1	0
(0x4005_0040)	bit Symbol		TMPREG5		
	Read/Write		R/W		
	After reset		0x00000000		
	Function				

<TMPREG0> Temporary register 0
<TMPREG1> Temporary register 1
<TMPREG2> Temporary register 2
<TMPREG3> Temporary register 3
<TMPREG4> Temporary register 4
<TMPREG5> Temporary register 5

16.4.2 Common Registers

16.4.2.1 VETADC Register

VETADC
(0x4005_0178)

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			

	15	. . .	1	0
bit Symbol	TADC			
Read/Write	R/W			
After reset	0x0000			
Function	ADC conversion time 0x0000-0xFFFF			

<TADC> The value to be set is:
ADC conversion time [s] ÷ PWM counter clock frequency [s]

* This register is effective when the 1-shunt current detection mode is selected and PWM shift is enabled.

16.4.3 Channel-Specific Registers

16.4.3.1   VEMODE Register

VEMODE1 (0x4005_00E0)		31			...			9	8
	bit Symbol								
	Read/Write				R				
	After reset				0x000000				
	Function				−				

		7	6	5	4	3	2	1	0
	bit Symbol		−			OCRMD		ZIEN	PVIEN
	Read/Write		R/W			R/W		R/W	R/W
	After reset		0			00		0	0
	Function	Write as '0'				Output control 00: Output OFF 01: Output enable 10: Reserved 11: Output OFF and EMG return		Zero-current detection 0: Disable 1: Enable	Phase interpolation 0: Disable 1: Enable

<PVIEN> Enables or disables phase interpolation.
<ZIEN> Enables or disables zero-current detection.
<OCRMD> Controls output operation (output OFF, output enable or EMG return).

16.4.3.2 VEFMODE Register

		31	. . .					17	16
VEFMODE1	bit Symbol								
(0x4005_00E4)	Read/Write				R				
	After reset				0x0000				
	Function				−				

		15	14	13	12	11	10	9	8
	bit Symbol	−						MREGDIS	−
	Read/Write				R/W			R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	Write as '0'						Keep the previous value of SIN/COS/ SECTOR 0: effective 1:no_effective	Write as '0'

		7	6	5	4	3	2	1	0
	bit Symbol	−	−	−	−	IDMODE		SPWMEN	C2PEN
	Read/Write		R/W			R/W		R/W	R/W
	After reset		0000			00		0	0
	Function	Always write as "0101".				Current detection mode 00: 3-shunt 01: 2-sensor 10: 1-shunt (for up count PMDTRG) 11: 1-shunt (for down count PMDTRG)		PWM shift enable 0: Disable 1: Enable	Modulation mode 0: 3-phase modulation 1: 2-phase modulation

<C2PEN>: Selects 3-phase or 2-phase modulation.
<SPWMEN>: Enables or disables PWM shift.
<IDMODE>: Selects the 3-shunt, 2-sensor or 1-shunt current detection mode.

When the 1-shunt mode is used, the acceptable PMDTRG is as follows.

VEFMODE1 <IDMODE>	PMD1TRGCR <TRG0MD>	PMD1TRGCR <TRG1MD>
10	010 (up-count)	010 (up-count)
10	101 (carrier bottom)	010 (up-count)
11	001 (down-count)	001 (down-count)
11	001 (down-count)	101 (carrier bottom)

<MREGDIS>: Keep or Not_keep the previous value of SIN/COS/SECTOR.
* In case of no_effective, SINM=SIN,COSM=COS,SECTORM=SECTOR.

16.4.3.3 VETPWM Register

VETPWM1 (0x4005_00E8)		31	· · ·	17	16
	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	TPWM			
	Read/Write	R/W			
	After reset	0x0000			
	Function	PWM period rate 0x0000 to 0xFFFF			

<TPWM> Set a PWM period rate (it is valid when the phase interpolation is enabled,
16-bit fixed-point data: 0.0 to 1.0) as follows:
PWM period [s] × Max_Hz×2^16
* Max_Hz: Maximum rotation speed
* It indicates a ratio between PWM frequency and maximum rotation speed.

16.4.3.4 VEOMEGA Register

VEOMEGA1 (0x4005_00EC)		31	· · ·	17	16
	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	OMEGA			
	Read/Write	R/W			
	After reset	0x0000			
	Function	Rotation speed 0x0000 to 0xFFFF			

<OMEGA> Set a rotation speed (16-bit fixed-point data: -1.0 to 1.0) as follows:
Rotation speed [Hz] ÷ Max_Hz×2^15

* Max_Hz: Maximum rotation speed [Hz]

16.4.3.5 VETHETA Register

VETHETA1 (0x4005_00F0)		31	· · ·	17	16
	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	THETA			
	Read/Write	R/W			
	After reset	0x0000			
	Function	Phase data			

<THETA> Set phase data (16-bit fixed-point data: 0.0 to 1.0) as follows:
Phase [deg] ÷ 360 × 2^16

16.4.3.6   VESIN/COS Registers

VECOS1
(0x4005_0138)

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			
	15	. . .	1	0
bit Symbol	COS			
Read/Write	R/W			
After reset	0x0000			
Function	Cosine value: 0x0000-0xFFFF			

VESIN1
(0x4005_013C)

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			
	15	. . .	1	0
bit Symbol	SIN			
Read/Write	R/W			
After reset	0x0000			
Function	Sine value: 0x0000-0xFFFF			

VECOSM1
(0x4005_0140)

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			
	15	. . .	1	0
bit Symbol	COSM			
Read/Write	R/W			
After reset	0x0000			
Function	Cosine value (previous value): 0x0000-0xFFFF			

VESINM1
(0x4005_0144)

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			
	15	. . .	1	0
bit Symbol	SINM			
Read/Write	R/W			
After reset	0x0000			
Function	Sine value (previous value): 0x0000-0xFFFF			

<COS> Cosine value based on the THETA value (16-bit fixed-point data: -1.0 to 1.0)
<SIN>   Sine value based on the THETA value (16-bit fixed-point data: -1.0 to 1.0)
<COSM> Previous value of the COS register
<SINM>   Previous value of the SIN register

16.4.3.7 dq Current Reference Registers

		31	· · ·	17	16
VEIDREF1	bit Symbol				
(0x4005_00F4)	Read/Write		R		
	After reset		0x0000		
	Function		−		
		15	· · ·	1	0
	bit Symbol		IDREF		
	Read/Write		R/W		
	After reset		0x0000		
	Function		Reference value of d-axis current: 0x0000 to 0xFFFF		

		31	· · ·	17	16
VEIQREF1	bit Symbol				
(0x4005_00F8)	Read/Write		R		
	After reset		0x0000		
	Function		−		
		15	· · ·	1	0
	bit Symbol		IQREF		
	Read/Write		R/W		
	After reset		0x0000		
	Function		Reference value of q-axis current: 0x0000 to 0xFFFF		

<ID_REF> Reference value of d-axis current (16-bit fixed-point data: -1.0 to 1.0)
<IQ_REF> Reference value of q-axis current (16-bit fixed-point data: -1.0 to 1.0)

The value to be set is:
Axis current reference [A] ÷ Max_I×2^15

* Max_I: (Phase current value [A] which corresponds to 1 LSB of ADC) × 2^11

16.4.3.8 dq Voltage Registers

		31	· · ·	1	0
VEVD1	bit Symbol		VD		
(0x4005_00FC)	Read/Write		R/W		
	After reset		0x00000000		
	Function		d-axis voltage: 0x00000000-0xFFFFFFFF		

		31	· · ·	1	0
VEVQ1	bit Symbol		VQ		
(0x4005_0100)	Read/Write		R/W		
	After reset		0x00000000		
	Function		q-axis voltage: 0x00000000-0xFFFFFFFF		

<VD> d-axis voltage (32-bit fixed-point data: -1.0 to 1.0)
<VQ> q-axis voltage (32-bit fixed-point data: -1.0 to 1.0)

The value to be set is as follows:
Axis voltage ÷ Max_V×2^31

* Max_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC) × 2^12

16.4.3.9　PI Control Coefficient Registers

VECIDKI1		31	· · ·	17	16
(0x4005_0104)	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	CIDKI			
	Read/Write	R/W			
	After reset	0x0000			
	Function	Integral coefficient for PI control of d-axis: 0x0000-0xFFFF			

VECIDKP1		31	· · ·	17	16
(0x4005_0108)	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	CIDKP			
	Read/Write	R/W			
	After reset	0x0000			
	Function	Proportional coefficient for PI control of d-axis: 0x0000-0xFFFF			

VECIQKI1		31	· · ·	17	16
(0x4005_010C)	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	CIQKI			
	Read/Write	R/W			
	After reset	0x0000			
	Function	Integral coefficient for PI control of q-axis: 0x0000-0xFFFF			

VECIQKP1		31	· · ·	17	16
(0x4005_0110)	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	−			
		15	· · ·	1	0
	bit Symbol	CIQKP			
	Read/Write	R/W			
	After reset	0x0000			
	Function	Proportional coefficient for PI control of q-axis: 0x0000-0xFFFF			

<CIDKI> Integral coefficient for PI control of d-axis
<CIDKP> Proportional coefficient for PI control of d-axis
<CIQKI> Integral coefficient for PI control of q-axis
<CIQKP> Proportional coefficient for PI control of q-axis

16.4.3.10 PI Control Integral Term Registers

VEVD_H1 (0x4005_0114)		31	· · ·	1	0
	bit Symbol	VDIH			
	Read/Write	R/W			
	After reset	0x00000000			
	Function	Upper 32 bits of VDI			

VEVDILH1 (0x4005_0118)		31	· · ·	17	16
	bit Symbol	VDILH			
	Read/Write	R/W			
	After reset	0x0000			
	Function				
		15	· · ·	1	0
	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	—			

VEVQIH1 (0x4005_011C)		31	· · ·	1	0
	bit Symbol	VQIH			
	Read/Write	R/W			
	After reset	0x00000000			
	Function				

VEVQILH1 (0x4005_0120)		31	· · ·	17	16
	bit Symbol	VQILH			
	Read/Write	R/W			
	After reset	0x0000			
	Function				
		15	· · ·	1	0
	bit Symbol				
	Read/Write	R			
	After reset	0x0000			
	Function	—			

<VDIH> Upper 32 bits of the integral term (VDI) for PI control of d-axis
<VDILH> Bit 31 to 16 of the integral term (VDI) for PI control of d-axis

VDI: 64-bit fixed-point data with 63 fractional bits (-1.0 to 1.0)

<VQIH> Upper 32 bits of the integral term (VQI) for PI control of q-axis
<VQILH> Bit 31 to 16 of the integral term (VQI) for PI control of q-axis

VQI: 64-bit fixed-point data with 63 fractional bits (-1.0 to 1.0)

16.4.3.11   VEMCTLF Register

		31	· · ·					9	8
VEMCTLF1	bit Symbol								
(0x4005_00DC)	Read/Write				R				
	After reset				0x000000				
	Function				−				
		7	6	5	4	3	2	1	0
	bit Symbol	−	-	PLSLFM	PLSLF	-	LVTF	LAVFM	LAVF
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	Write as '0'	Write as '0'	Previous value of PLSLFM	Disparity of pulse width small flag	Write as '0'	Supply voltage lower flag	Previous LAVF value	Low-speed flag 0: High-speed 1: Low-speed

<LAVF> Low-speed flag   OMEGA $\geq$ FPWMCHG case = 0
                                              OMEGA $<$ FPWMCHG case =1
<LAVFM> Previous LAVF value
<LVTF> Supply voltage lower flag   VDC$\geqq$0x0100(1/128) case =0,
                                                             VDC$<$0x0100(1/128) case =1
＜PLSLF＞   Minimum disparity of pulse width$\geqq$MINPLS case =0
                     Minimum disparity of pulse width＜MINPLS case =1
<PLSLFM> Previous value of PLSLF

16.4.3.12   VEFPWMCHG Register

		31	· · ·		17	16
VEFPWMCHG1	bit Symbol					
(0x4005_0124)	Read/Write			R		
	After reset			0x0000		
	Function			−		
		15	· · ·		1	0
	bit Symbol			FPWMCHG		
	Read/Write			R/W		
	After reset			0x0000		
	Function			Rotation speed when PWM shift is enabled		

<FPWMCHG> Rortation speed when PWM shift is enabled. The value to be set is:
                    Rortation speed [Hz] $\div$ Max_Hz$\times 2^{15}$

                    * Max_Hz: Maximum rotation speed [Hz]

16.4.3.13　PWM Period Register

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			

	15	· · ·	1	0
bit Symbol	VMDPRD			
Read/Write	R/W			
After reset	0x0000			
Function	PWM period			

VEVMDPRD1 (0x4005_0128)

<VMDPRD> PWM period. Set the value of the PMD's MDPRD register.

16.4.3.14　VEMINPLS Register

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	−			

	15	· · ·	1	0
bit Symbol	MINPLS			
Read/Write	R/W			
After reset	0x0000			
Function	Minimum disparity of pulse width			

VEMINPLS1 (0x4005_012C)

<MINPLS>　Set the minimum disparity of pulse width　among the duty of VECMPU, VECMPV, VECMPW.

　Disparity of pulse width [s] ÷ PWM counter clock period [s]

16.4.3.15 Sector Registers

VESECTOR1
(0x4005_0148)

	31	· · ·						9	8
bit Symbol									
Read/Write	R								
After reset	0x000000								
Function	—								
	7	6	5	4	3	2	1	0
bit Symbol					SECTOR			
Read/Write	R				R/W			
After reset	0x0				0x0			
Function	—				Sector information 0x0-0xF:			

VESECTORM1
(0x4005_014C)

	31	· · ·						9	8
bit Symbol									
Read/Write	R								
Function	—								
After reset	0x000000								
	7	6	5	4	3	2	1	0
bit Symbol					SECTORM			
Read/Write	R				R/W			
After reset	0x0				0x0			
Function	—				Sector information 0x0-0xF:			

<SECTOR> Sector information. Indicates the rotation position at the time of output by 12 sectors each having 30 degrees.

<SECTORM> Previous sector information. Used in input processing.

16.4.3.16 Zero-Current Registers

VEIA01
(0x4005_0150)

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	—			
	15	· · ·	1	0
bit Symbol	IA0			
Read/Write	R/W			
After reset	0x0000			
Function	AD conversion result of U-phase at zero-current			

VEIB01
(0x4005_0154)

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	—			
	15	· · ·	1	0
bit Symbol	IB0			
Read/Write	R/W			
After reset	0x0000			
Function	AD conversion result of V-phase at zero-current			

VEIC01
(0x4005_0158)

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	—			
	15	· · ·	1	0
bit Symbol	IC0			
Read/Write	R/W			
After reset	0x0000			
Function	AD conversion result of W-phase at zero-current			

<IA0> Stores the AD conversion result of U-phase current when the motor is at stop.
<IB0> Stores the AD conversion result of V-phase current when the motor is at stop
<IC0> Stores the AD conversion result of W-phase current when the motor is at stop

   * When the zero-current detection is enabled, AD conversion results are automatically
     stored to these registers.
   * AD conversion results are stored in the 15--4 bits, with the 3--0 bits always "0".

16.4.3.17 Current ADC Result Registers

VEIAADC1 (0x4005_015C)		31	· · ·	17	16
	bit Symbol				
	Read/Write		R		
	After reset		0x0000		
	Function		−		
		15	· · ·	1	0
	bit Symbol		IAADC		
	Read/Write		R/W		
	After reset		0x0000		
	Function		AD conversion result of U-phase current : 0x0000-0xFFFF		

VEIBADC1 (0x4005_0160)		31	· · ·	17	16
	bit Symbol				
	Read/Write		R		
	After reset		0x0000		
	Function		−		
		15	· · ·	1	0
	bit Symbol		IBADC		
	Read/Write		R/W		
	After reset		0x0000		
	Function		AD conversion result of V-phase current: 0x0000-0xFFFF		

VEICADC1 (0x4005_0164)		31	· · ·	17	16
	bit Symbol				
	Read/Write		R		
	After reset		0x0000		
	Function		−		
		15	· · ·	1	0
	bit Symbol		ICADC		
	Read/Write		R/W		
	After reset		0x0000		
	Function		AD conversion result of W-phase current: 0x0000-0xFFFF		

<IAADC> Stores the AD conversion result of U-phase current.
<IBADC> Stores the AD conversion result of V-phase current.
<ICADC> Stores the AD conversion result of W-phase current.

* AD conversion results are stored in the 15--4 bits, with the 3--0 bits always "0".

16.4.3.18 Supply Voltage Register

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	—			

	15	· · ·	1	0
bit Symbol	VDC			
Read/Write	R/W			
After reset	0x0000			
Function	Supply voltage: 0x0000-0xFFFF			

VEVDC1
(0x4005_0168)

<VDC> Supply voltage (16-bit fixed-point data: 0 to 1.0)
   The actual voltage value is:
      VDC value ÷ Max_V value × 2^15

   * Max_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC) × 2^12

16.4.3.19 dq Current Registers

	31	· · ·	1	0
bit Symbol	ID			
Read/Write	R/W			
After reset	0x00000000			
Function	d-axis current: 0x00000000-0xFFFFFFFF			

VEID1
(0x4005_016C)

	31	· · ·	1	0
bit Symbol	IQ			
Read/Write	R/W			
After reset	0x00000000			
Function	q-axis current: 0x00000000-0xFFFFFFFF			

VEIQ1
(0x4005_0170)

<ID> d-axis current (32-bit fixed-point data: -1.0 to 1.0)
<IQ> q-axis current (32-bit fixed-point data: -1.0 to 1.0)

   The actual current value is:
      ID (or IQ) value ÷ Max_I value x 2^31

   * Max_I: (Phase current value [A] which corresponds to 1 LSB of ADC) × 2^11

16.4.3.20   PWM Duty Register

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	―			
	15	· · ·	1	0
bit Symbol	VCMPU			
Read/Write	R/W			
After reset	0x0000			
Function	PWM pulse width of U-phase: 0-0xFFFF			

VECMPU1 (0x4005_019C)

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	―			
	15	· · ·	1	0
bit Symbol	VCMPV			
Read/Write	R/W			
After reset	0x0000			
Function	PWM pulse width of V-phase: 0-0xFFFF			

VECMPV1 (0x4005_01A0)

	31	· · ·	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	―			
	15	· · ·	1	0
bit Symbol	VCMPW			
Read/Write	R/W			
After reset	0x0000			
Function	PWM pulse width of W-phase: 0-0xFFFF			

VECMPW1 (0x4005_01A4)

<VCMPU> PMD setting: PWM setting of U-phase
<VCMPV> PMD setting: PWM setting of V-phase
<VCMPW> PMD setting: PWM setting of W-phase

16.4.3.21 6-Phase Output Control Register

		31				...			17	16
VEOUTCR1 (0x4005_01A8)	bit Symbol									
	Read/Write					R				
	After reset					0x0000				
	Function					—				

		15	14	13	12	11	10	9	8
	bit Symbol								WPWM
	Read/Write					R			R/W
	After reset				0000000				0
	Function				—				PWM of W-phase 0: ON/OFF output 1: PWM output

		7	6	5	4	3	2	1	0
	bit Symbol	VPWM	UPWM	WOC		VOC		UOC	
	Read/Write	R/W	R/W	R/W		R/W		R/W	
	After reset	0	0	00		00		00	
	Function	PWM of V-phase 0: ON/OFF output 1: PWM output	PWM of U-phase 0: ON/OFF output 1: PWM output	Output control of W-phase 00:W0 OFF, Z0 OFF[*] 01: W0 ON, Z0 OFF 10: W0 OFF, Z0 ON 11: W0 ON, Z0 ON *W0 and Z0 are both ON when WPWM=1.		Output control of V-phase 00: V0 OFF, Y0 OFF[*] 01: V0 ON, Y0 OFF 10: V0 OFF, Y0 ON 11: V0 ON, Y0 ON *V0 and Y0 are both ON when VPWM=1.		Output control of U-phase 00: U0 OFF, X0 OFF[*] 01: U0 ON, X0 OFF 10: U0 OFF, X0 ON 11: U0 ON, X0 ON *U0 and X0 are both ON when UPWM=1.	

<UPWM>, <U0C> PMD setting: Output control of U-phase (U0,X0)

Setting		Output	
UPWM	U0C	U0	X0
0	00	OFF output	OFF output
1	00	PWMU inverted output	PWMU output
1	11	PWMU output	PWMU inverted output

*The table shows only those combinations that are used in the VE.

<VPWM>,<V0C> PMD setting: Output control of V-phase (V0,Y0)

Setting		Output	
VPWM	V0C	V0	Y0
0	00	OFF output	OFF output
1	00	PWMV inverted output	PWMV output
1	11	PWMV output	PWMV inverted output

*The table shows only those combinations that are used in the VE.

<WPWM>,<W0C> PMD setting: Output control of W-phase (W0,Z0)

Setting		Output	
WPWM	W0C	W0	Z0
0	00	OFF output	OFF output
1	00	PWMW inverted output	PWMW output
1	11	PWMW output	PWMW inverted output

* The table shows only those combinations that are used in the VE.

16.4.3.22 VETRGCRC Register

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	–			

	15	. . .	1	0
bit Symbol	TRGCRC			
Read/Write	R/W			
After reset	0x0000			
Function	Trigger correction			

VETRGCRC1
(0x4005_0130)

<TRGCRC>: Used to correct the synchronizing trigger timing. The value to be set is:
Correction time [s] ÷ PWM counter clock frequency [s]

16.4.3.23 VETRGCMP Register

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	–			

	15	. . .	1	0
bit Symbol	VTRGCMP0			
Read/Write	R/W			
After reset	0x0000			
Function	Trigger timing setting 0x0000: Prohibited 0x0001 to (MDPRD value -1): Trigger timing MDPRD value to 0xFFFF: Prohibited			

VETRGCMP01
(0x4005_01AC)

	31	. . .	17	16
bit Symbol				
Read/Write	R			
After reset	0x0000			
Function	–			

	15	. . .	1	0
bit Symbol	VTRGCMP1			
Read/Write	R/W			
After reset	0x0000			
Function	Trigger timing setting 0x0000:Prohibited 0x0001 to (MDPRD value-1): Trigger timing MDPRD value to 0xFFFF: Prohibited			

VETRGCMP11
(0x4005_01B0)

<VTRGCMP0> PMD setting: Specifies the trigger timing for sampling ADC in synchronization with PMD.

<VTRGCMP1> PMD setting: Specifies the trigger timing for sampling ADC in synchronization with PMD.

* These registers are effective when one of the following PMD trigger modes is selected:
count-down match, count-up match, count-up/-down match

* These registers are ineffective when the PMD trigger output mode is set to trigger select output (TRGOUT=1).

16.4.3.24　VETRGSEL Register

		31	· · ·						9	8
VETRGSEL1	bit Symbol									
(0x4005_01B4)	Read/Write					R					
	After reset					0x000000					
	Function					—					

		7	6	5	4	3	2	1	0
	bit Symbol							VTRGSEL	
	Read/Write			R				R/W	
	After reset			00000				000	
	Function			—			Synchronizing trigger 0-5: Output trigger number 6-7: Prohibited		

<VTRGSEL> PMD setting: Specifies the synchronizing trigger number to be output at the timing specified in the VTRGCMP0 register.

* These registers are effective when the PMD trigger output mode is set to trigger select output (TRGOUT= 1).

16.4.3.25　VEEMGRS Register

		31	· · ·				9	8
VEEMGRS1	bit Symbol							
(0x4005_01B8)	Read/Write			R				
	After reset			0x000000				
	Function			—				

		7	· · ·	2	1	0
	bit Symbol					EMGRS
	Read/Write		R			R/W
	After reset		0000000			0
	Function		—			EMG return 0: Nop 1: EMG return command

<EMGRS> PMD setting : EMG return command for returning from the EMG state

16.5 Description of Operations

16.5.1 Schedule Management

Figure 16-4 shows a flowchart for motor control. The Vector Engine makes state transitions according to the schedule and mode settings which are programmed through the relevant registers.
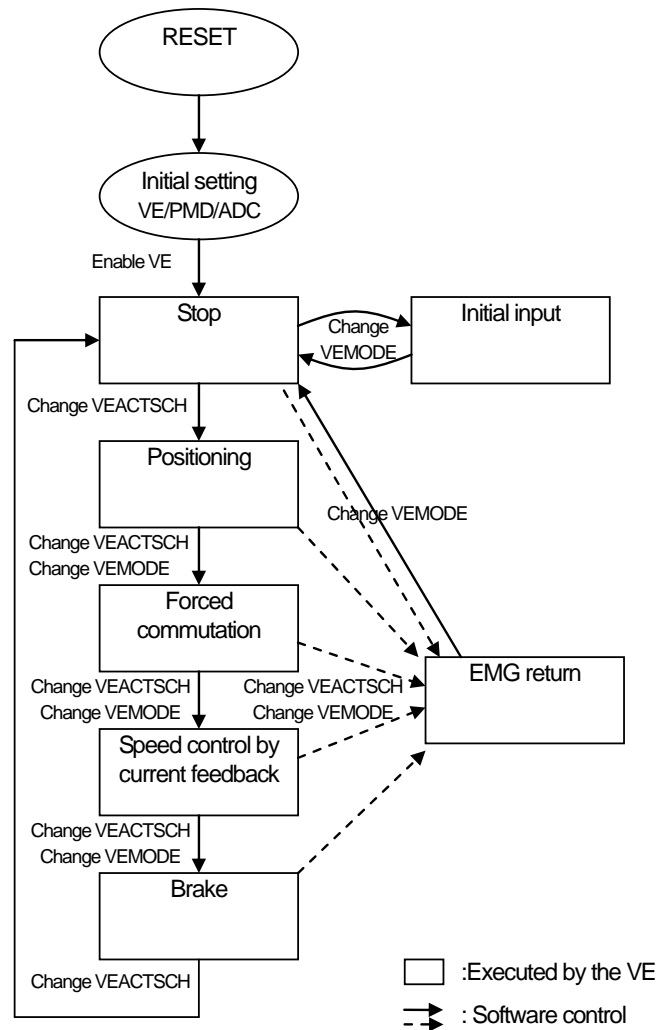
Figure 16-4   Example of Motor Control Flow

```
RESET               : Microcontroller reset
Initial setting     : Initial setting by a user-created program
Stop                : Stop the motor.
Initial input       : Sample and store zero-current data when the motor is at stop
Positioning         : Determine the initial motor position.
Forced commutation  : Start the motor. For a specified period, the motor is rotated at a specified
                      speed, not controlled by current feedback.
Speed control by
 current feedback    : Control motor rotation by current feedback.
Brake               : Deceleration control
EMG return          : Return from the EMG state.
```

16.5.1.1   Schedule Control

The VEACTSCH register is used to select the schedule to be executed. A schedule is comprised of an output schedule handling output-related tasks and an input schedule handling input-related tasks. Table 16-3 shows the tasks that are executed in each schedule.

The VEMODE register is used to enable or disable phase interpolation, control output operation, and enable or disable zero-current detection as appropriate for each step of the motor control flow (see Table 16-4).

Table 16-3   Tasks To Be Executed in Each Schedule

Schedule Selection (VEACTSCH)	Output Schedule						Input Schedule		
	Current control	SIN/COS computation	Output coordinate axis conversion	Output phase conversion	Output control	Trigger generation	Input processing	Input phase conversion	Input coordinate axis conversion
0 Individual execution	*1	*1	*1	*1	*1	*1	*1	*1	*1
1 Schedule 1	✔	✔*2	✔	✔	✔*3	✔	✔*4	✔	✔
4 Schedule 4	–	✔*2	✔	✔	✔*3	✔	✔*4	✔	✔
9 Schedule 9	–	–	–	–	✔*3	✔	✔*4	–	–

*1: Each task is executed only when it is specified.

*2: Phase interpolation

*3: Output OFF/EMGRS

*4: Task operation to be switched by zero-current detection

Table 16-4   Typical Setting Example

Motor Control Flow	Register Setting				
	Schedule selection (VEACTSCH)	Task specification (VETASKAPP)	Phase interpolation (VEMODE)	Output control (VEMODE)	Zero-current detection (VEMODE)
Stop	9	0	x	00	0
Initial input	9	0	x	00	1
Positioning	1	5	0	01	0
Forced commutation	1	5	1	01	0
Speed control by current feedback	1	5	1	01	0
Brake	4	6	0	01	0
EMG return	9	0	x	11	0

An output schedule begins executing by the VECPURUNTRG command. When all output-related tasks are completed, the Vector Engine enters a standby state and waits for a start trigger for input-related tasks. At this time, schedules of the other channel can be executed.

An input schedule begins executing by a start trigger. When all input-related tasks are completed, the Vector Engine generates an interrupt to the CPU and enters a halt state. However, if the schedule has its repeat count (VEREPTIME) set to "2" or more, an interrupt is not generated until the schedule is executed the specified number of times.
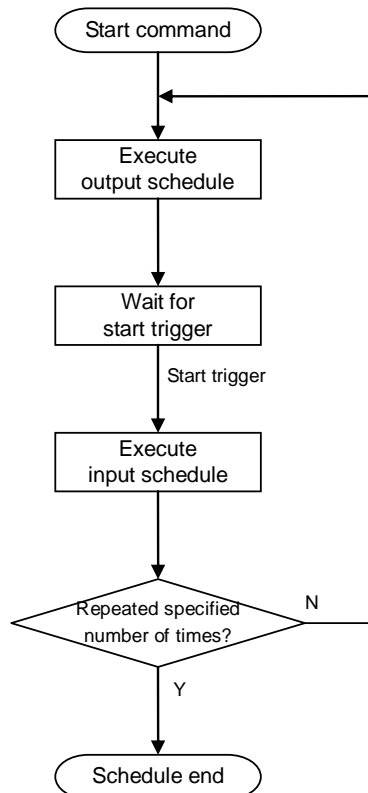
Figure 16-5　Schedule Execution Flow

16.5.1.2   Start Control

Enable the Vector Engine with the VEEN register. Specify a schedule (VEACTSCH register), task to be executed (VETASKAPP register) and repeat count (VEREPTIME register).

A schedule of the Vector Engine is comprised of an output schedule and an input schedule. Typically, the Vector Engine executes an output schedule first, enters a standby state, and then starts executing an input schedule by a start trigger.

An output schedule is started:

i) By the VECPURUNTRG command. In this case, the task specified in the VETASKAPP register is executed.
ii) On a repeat start (when VEREPTIME $\geq$ 2) after the corresponding input schedule is completed.

An input schedule is started:

i) By a start trigger (selected in the VETRGMODE register) after the corresponding output schedule is completed.
ii) By the VECPURUNTRG command. In this case, the task specified in the VETASKAPP register is executed.

16.5.2 Summary of Tasks

Table 16-5 gives a summary of tasks executed in output and input schedules. When each task is to be executed individually or specified as a startup task, use the task number shown in this table.

Table 16-5 List of Tasks

Task		Description	Task Number
Output schedule	Current control	Controls dq currents.	5
	SIN/COS computation	Performs sine/cosine computation and phase interpolation.	6
	Output coordinate axis conversion	Converts dq coordinates to αβ coordinates.	7
	Output phase conversion	Converts 2-phase to 3-phase.	8
	Output control	Converts data to PMD setting format. Switches PWM shift.	0
	Trigger generation	Generates synchronization trigger timing.	1
Input schedule	Input processing	Captures AD conversion results and converts them into fixed-point format	2
	Input phase conversion	Converts 2-phase to 3-phase.	3
	Input coordinate axis conversion	Converts αβ coordinates to dq coordinates.	4

16.5.2.1   Current Control

The current control unit is comprised of a PI control unit for d-axis and a PI control unit for q-axis, and calculates d-axis and q-axis voltages.

i) PI control of d-axis current

[Equations]

$\Delta ID = VEIDREF \times 2^{16} - ID$: Difference between current reference value and current feedback

$VDI = VECIDKI \times 2^{16} \times \Delta ID \times 2 + VDI$: Integral term computation

$VEVD = (VECIDKP \times 2^{16} \times \Delta ID \times 2 + VDI) / 2^{32}$: Voltage calculation using proportional term

	Register Name	Description	
Input	VEID1	d-axis current	32-bit fixed-point data (31 fractional bits)
	VEIDREF1	Reference value of d-axis current	16-bit fixed-point data (15 fractional bits)
	VECIDKP1	Proportional coefficient	16-bit data
	VECIDKI1	Integral coefficient	16-bit data
Output	VEVD1	d-axis voltage	32-bit fixed-point data (31 fractional bits)
Internal	VDI1	Integral term of d-axis voltage	64-bit fixed-point data (63 fractional bits)

ii) PI control of q-axis current

[Equations]

$\Delta IQ = VEIQREF \times 2^{16} - IQ$: Difference between current reference value and current feedback

$VQI = VECIQKI \times 2^{16} \times \Delta IQ \times 2 + VQI$: Integral term computation

$VEVQ = (VECIQKP \times 2^{16} \times \Delta IQ \times 2 + VQI) / 2^{32}$: Voltage calculation using proportional term

	Register Name	Description	
Input	VEIQ1	q-axis current	32-bit fixed-point data (31 fractional bits)
	VEIQREF1	Reference value of q-axis current	16-bit fixed-point data (15 fractional bits)
	VECIQKP1	Proportional coefficient	16-bit data
	VECIQKI1	Integral coefficient	16-bit data
Output	VEVQ1	q-axis voltage	32-bit fixed-point data (31 fractional bits)
Internal	VQI1	Integral term of q-axis voltage	64-bit fixed-point data (63 fractional bits)

16.5.2.2 SIN/COS Computation

The SIN/COS computation unit is comprised of a phase interpolation unit and a SIN/COS computation unit.

Phase interpolation calculates the rotation speed by integrating with the PWM period. It is executed only when phase interpolation is enabled.

i) Phase interpolation
[Equations]
VETHEATA = (VEOMEGA × VETPWM + VETHEATA × 2^31) / 2^31   : Integration of rotation speed.
Only when phase interpolation is enabled.

	Register Name	Description	
Input	VETHETA1	Phase θ	16-bit fixed-point data (0.0 to1.0, 16 fractional bits)
	VEOMEGA1	Rotation speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VETPWM	PWM period rate	16-bit data
	VEMODE1	Phase interpolation enable	Mode settings
Output	VETHETA1	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits))

ii) SIN/COS computation
[Equations]
VESINM = VESIN   : Saves previous value (for input processing).
VECOSM = VECOS   : Saves previous value (for input processing).
VESIN = sin(VETHETA × π／2^15) ×2^15   : SIN/COS computation
VECOS = sin((VETHETA＋0x4000)×π／2^15) ×2^15

	Register Name	Description	
Input	VETHETA1	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
Output	VESIN1	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOS1	Cosine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VESINM1	Previous sine value	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOSM1	Previous cosine value	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)

16.5.2.3   Output Voltage Conversion (Coordinate axis Conversion/Phase Conversion)

Output voltage conversion is comprised of dq-to-αβ coordinate axis conversion and 2-phase-to-3-phase conversion.

The dq-to-αβ coordinate axis conversion calculates Vα and Vβ from Vd, Vq in SIN and COS.

The 2-phase-to-3-phase conversion performs segmentation by using Vα and Vβ and performs space vector conversion to calculate Va, Vb and Vc.

For the 2-phase-to-3-phase conversion, either 2-phase modulation or 3-phase modulation can be selected.

(1) dq-to-αβ coordinate conversion
[Equations]

VETMPREG3 = (VECOS ×VEVD - VESIN ×VEVQ) / 2^15   : Calculates Vα.
VETMPREG4 = (VESIN ×VEVD + VECOS ×VEVQ) / 2^15   : Calculates Vβ.

	Register Name	Description	
Input	VEVD1	d-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEVQ1	q-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VESIN1	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOS1	Cosine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
Output	VETMPREG3	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)

(2) 2-phase-to-3-phase conversion (space vector conversion)
 i) Segmentation
[Equations]

```
VESECTORM = VESECTOR                    : Saves previous sector.
if(Vα≥0 & Vβ≥0)
  if(|Vα|≥|Vβ|÷√3)
    if(|Vα|÷√3≥|Vβ|)        SECTOR=0
    else                    SECTOR=1
  else                      SECTOR=2
else if(Vα<0 & Vβ≥0)
  if(|Vα|<|Vβ|÷√3)          SECTOR=3
  else if(|Vα|÷√3<|Vβ|)     SECTOR=4
    else                    SECTOR=5
else if(Vα<0 & Vβ<0)
  if(|Vα| ≥ |Vβ|÷√3)
    if(|Vα|÷√3≥|Vβ|)        SECTOR=6
    else                    SECTOR=7
  else                      SECTOR=8
else if(Vα≥0 & Vβ<0)
  if(|Vα| < |Vβ|÷√3)        SECTOR=9
  else if(|Vα|÷√3<|Vβ|)     SECTOR=10
    else                    SECTOR=11
```

	Register Name	Description	
Input	VETMPREG3	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
Output	VESECTOR1	Sector	4-bit data
	VESECTORM1	Previous sector	4-bit data

ii) 3-phase voltage calculation (when 3-phase modulation is selected and SECTOR=0)
    [Equations]

$t1 = \sqrt{3}/VEVDC \times 2^{15} \times (\sqrt{3}/2 \times V\alpha - 1/2 \times V\beta)$: Calculates V1 period.

$t2 = \sqrt{3}/VEVDC \times 2^{15} \times V\beta$: Calculates V2 period.

$t3 = 1 - t1 - t2$: Calculates V0+V7 period.

$VETMPREG0 = t1 + t2 + t3 \div 2$: Calculates Va.

$VETMPREG1 = t1 + t3 \div 2$: Calculates Vb.

$VETMPREG2 = t3 \div 2$: Calculates Vc.

	Register Name	Description	
Input	VETMPREG3	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEVDC1	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VESECTOR1	Sector	4-bit data
	VEFMODE1	Modulation mode	Mode settings
Output	VETMPREG0	a-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)

16.5.2.4   Output Control

The output control unit converts 3-phase voltage values into PWM setting format (VECMPU, VECMPV and VECMPW), and sets the VEOUTCR register to control output operation.

When 1-shunt current detection and 2-phase modulation are selected and PWM is enabled, if the rotation speed is slower than the PWM shift switching reference value, output is switched to shift PWM output.

	Register Name	Description	
Input	VETMPREG0	a-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VEMDPRD1	PWM period	16-bit data (PMD PWM period)
	VESECTOR1	Sector	4-bit data
	VEOMEGA1	Rotation speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEFPWMCHG1	PWM shift switching reference	16-bit fixed-point data (0 to 1.0, 15 fractional bits)
	VEMODE1	Output control operation	Mode settings
	VEFMODE1	shift enable/ modulation mode/ detection mode	Mode settings
Output	VECMPU1	PMD U-phase PMW setting	16- bit data (0 to MDPRD value)
	VECMPV1	PMD V-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPW1	PMD W-phase PWM setting	16-bit data (0 to MDPRD value)
	VEOUTC1R1	PMD output control setting	9-bit setting
	VEEMGRS1	PMD EMG return	1-bit setting
	VEMCTLF1	Shift switching flag	Status

16.5.2.5 Trigger Generation

The trigger generation unit calculates the trigger timing from the PWM setting values (VECMPU, VECMPV and VECMPW) as appropriate to the current detection method, and sets the VETRGCMP0 and VETRGCMP1 registers.

	Register Name	Description	
Input	VECMPU1	PMD U-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPV1	PMD V-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPW1	PMD W-phase PWM setting	16- bit data (0 to MDPRD value)
	VEMDPRD1	PWM period setting	16-bit data (PMD PWM period)
	VETADC	AD conversion time	16-bit data (0 to MDPRD value)
	VETRGCRC1	Trigger correction value	16-bit data (0 to MDPRD value)
	VESECTOR1	Sector	4-bit data
	VEMODE1	Output control operation	Mode settings
	VEFMODE1	shift enable/ modulation mode/ detection mode	Mode settings
	VEMCTLF1	Shift switching flag	Status
Output	VETRGCMP0	PMD trigger 0 timing	16-bit data (0 to MDPRD value)
	VETRGCMP1	PMD trigger 1 timing	16-bit data (0 to MDPRD value)
	VETRGSEL1	PMD trigger selection	3-bit data

16.5.2.6   Input Processing

The input processing unit saves segmented 3-phase current conversion results, and converts the current and voltage conversion results into fixed-point data.
It saves zero-current conversion results in the initial input processing..

	Register Name	Description	
Input	VEADREG0A	ADC unit A conversion result 0	16-bit data (The upper 12 bits are used.)
	VEADREG1A	ADC unit A conversion result 1	16-bit data (The upper 12 bits are used.)
	VEADREG2A	ADC unit A conversion result 2	16-bit data (The upper 12 bits are used.)
	VEADREG3A	ADC unit A conversion result 3	16-bit data (The upper 12 bits are used.)
	VEADREG0B	ADC unit B conversion result 0	16-bit data (The upper 12 bits are used.)
	VEADREG1B	ADC unit B conversion result 1	16-bit data (The upper 12 bits are used.)
	VEADREG2B	ADC unit B conversion result 2	16-bit data (The upper 12 bits are used.)
	VEADREG3B	ADC unit B conversion result 3	16-bit data (The upper 12 bits are used.)
	VEPHNUM0A	ADREG0A detected phase information	2-bit data
	VEPHNUM1A	ADREG1A detected phase information	2-bit data
	VEPHNUM2A	ADREG2A detected phase information	2-bit data
	VEPHNUM3A	ADREG3A detected phase information	2-bit data
	VEPHNUM0B	ADREG0B detected phase information	2-bit data
	VEPHNUM1B	ADREG1B detected phase information	2-bit data
	VEPHNUM2B	ADREG2B detected phase information	2-bit data
	VEPHNUM3B	ADREG3B detected phase information	2-bit data
	VESECTORM1	Sector information	4-bit data
	VEMODE1	Zero-current detection	Mode settings
	VEFMODE1	PMD channel /current detection mode / ADC unit /shift enable	Mode settings
	VEMCTLF1	Shift switching flag	Status
Output	VEVDC1	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VETMPREG0	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
Internal	IA01	a-phase zero-current conversion result	16-bit data (The upper 12 bits are used.)
	IB01	b-phase zero-current conversion result	16-bit data (The upper 12 bits are used.)
	IC01	c-phase zero-current conversion result	16-bit data (The upper 12 bits are used.)
	IAADC1	a-phase current conversion result	16-bit data (The upper 12 bits are used.)
	IBADC1	b-phase current conversion result	16-bit data (The upper 12 bits are used.)
	ICADC1	c-phase current conversion result	16-bit data (The upper 12 bits are used.)

16.5.2.7   Input Current Conversion (Phase Conversion/Coordinate axis Conversion)

Input current conversion is comprised of 3-phase-to-2-phase conversion and αβ-to-dq coordinate axis conversion.

The 3-phase-to-2-phase conversion calculates Iα and Iβ from Ia, Ib and Ic.

The αβ-to-dq coordinate axis conversion calculates Id and Iq from Iα, Iβ, VESINM and VECOSM.

(1) 3-phase-to-2-phase conversion

[Equations]

VETMPREG3 = VETMPREG0   : Calculates Iα.

VETMPREG4 = 1/√3 ×VETMPREG1 - 1/√3 ×VETMPREG2   : Calculates Iβ.

	Register Name	Description	
Input	VETMPREG0	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
Output	VETMPREG3	α-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)

(2) αβ-to-dq coordinate conversion

[Equations]

VEID = VECOSM×VETMPREG3 + VESINM×VETMPREG4   : Calculates Id.

VEIQ = -VESINM×VETMPREG3 + VECOSM×VETMPREG4   : Calculates Iq.

	Register Name	Description	
Input	VETMPREG3	α-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	B-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VESINM1	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOSM1	Cosine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
Output	VEID1	d-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEIQ1	q-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)

16.6 Combinations of VE and ADC

The Vector Engine calculates the stored value in AD conversion result register 0 to 2 (ADBREG0 to 2) as a current data and calculates the sotred value in AD conversion result register 3 (ADBREG3) as a voltage data. Therefore, please specify it with proper setting, referring to Table 16-6.

Table 16-6 Combination of VE and ADC

Vector Engine	ADC			
Current detection <IDMODE>	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0x	Current data 1	Current data 2	Note	VDC data
1x	Current data 1	Current data 2	-	VDC data

Note : Specifying the phase information to the register is necessary. However the AD conversion result of its register is not used for calculation.

17. Motor Control Circuit (PMD: Programmable Motor Driver)

The TMPM372/3/4 contains one channel programmable motor driver (PMD). The PMD of this product has newly added features of conduction output control and DC overvoltage detection to realize sensorless motor control and supports interaction with the AD converter.
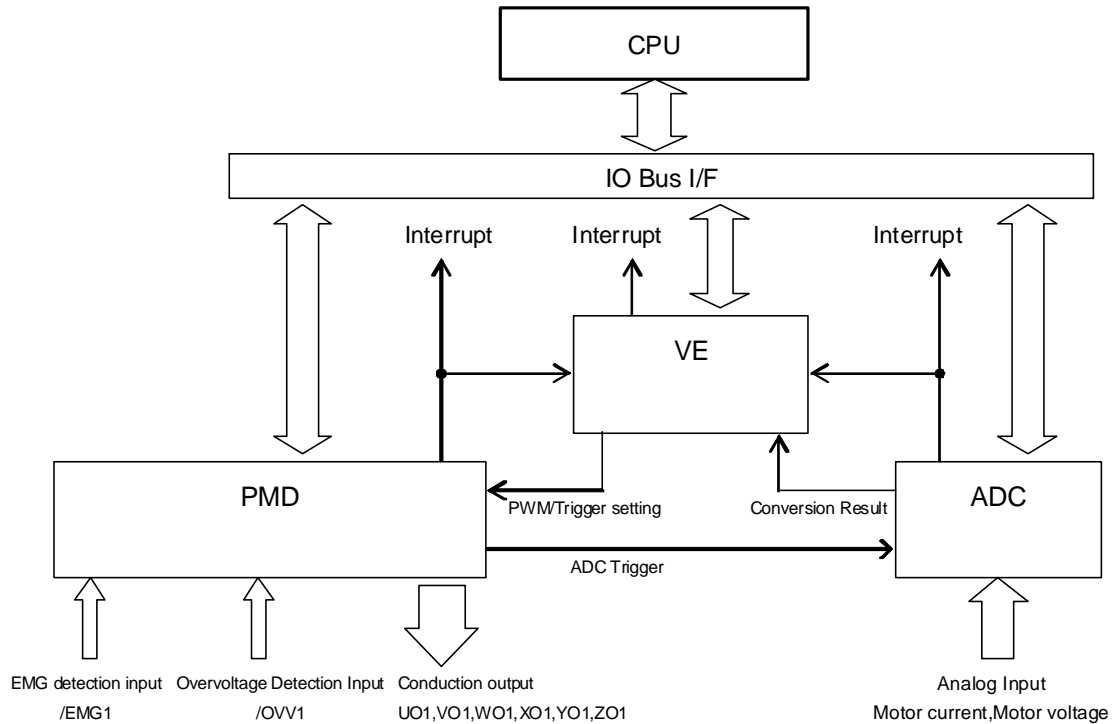
Figure 17-1 Motor Control-related Block Constitution

17.1 PMD Input/Output Signals

The table below shows the signals that are input to and output from PMD.

Table 17-1 Input/Output Signals

Channel	Pin Name	PMD Signal Name	Description
PMD1	PG7/$\overline{OVV1}$	OVV 1	OVV state signal
	PG6/$\overline{EMG1}$	EMG 1	EMG state signal
	PG0/UO1	UO 1	U-phase output
	PG1/XO1	X O1	X-phase output
	PG2/VO1	V O1	V-phase output
	PG3/YO1	Y O1	Y-phase output
	PG4/WO1	W O1	W-phase output
	PG5/ZO1	Z O1	Z-phase output

17.2 PMD Registers

The table below shows the registers related to the PMD.

Table 17-2 PMD Registers

Address	Mnemonic	Register Name
0x4005 0480	PMD1MDEN	PMD Enable Register
0x4005 0484	PMD1PORTMD	Port Output Mode Register
0x4005 0488	PMD1MDCR	PMD Control Register
0x4005 048C	PMD1CNTSTA	PWM Counter Status Register
0x4005 0490	PMD1MDCNT	PWM Counter Register
0x4005 0494	PMD1MDPRD	PWM Period Register
0x4005 0498	PMD1CMPU	PMD Compare U Register
0x4005 049C	PMD1CMPV	PMD Compare V Register
0x4005 04A0	PMD1CMPW	PMD Compare W Register
0x4005 04A4	PMD1MODESEL	Mode Select Register
0x4005 04A8	PMD1MDOUT	PMD Output Control Register
0x4005 04AC	PMD1MDPOT	PMD Output Setting Register
0x4005 04B0	PMD1EMGREL	EMG Release Register
0x4005 04B4	PMD1EMGCR	EMG Control Register
0x4005 04B8	PMD1EMGSTA	EMG Status Register
0x4005 04BC	PMD1OVVCR	OVV Control Register
0x4005 04C0	PMD1OVVSTA	OVV Status Register
0x4005 04C4	PMD1DTR	Dead Time Register
0x4005 04C8	PMD1TRGCMP0	Trigger Compare 0 Register
0x4005 04CC	PMD1TRGCMP1	Trigger Compare 1 Register
0x4005 04D0	PMD1TRGCMP2	Trigger Compare 2 Register
0x4005 04D4	PMD1TRGCMP3	Trigger Compare 3 Register
0x4005 04D8	PMD1TRGCR	Trigger Control Register
0x4005 04DC	PMD1TRGMD	Trigger Output Mode Setting Register
0x4005 04E0	PMD1TRGSEL	Trigger Output Select Register
0x4005 04FC	Reserved	-

Note: Do not access to "Reserved" address.

17.3 PMD Circuit

17.3.1 Configuration of PMD
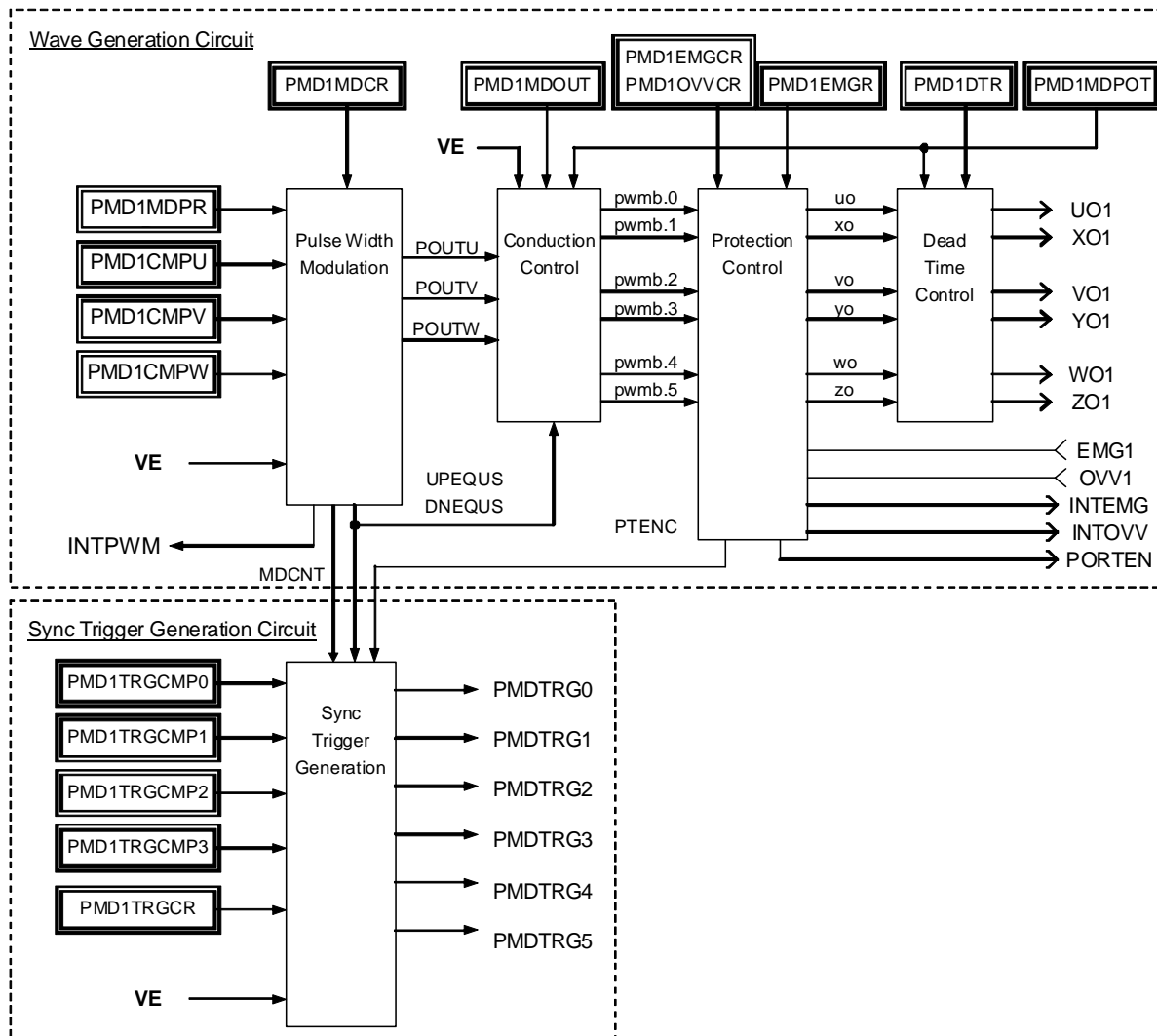
Figure 17-2 Block diagram of PMD Circuit

The PMD circuit consists of two blocks of a wave generation circuit and a sync trigger generation circuit. The wave generation circuit includes a pulse width modulation circuit, a conduction control circuit, a protection control circuit, a dead time control circuit.

- The pulse width modulation circuit generates independent 3-phase PWM waveforms with the same PWM frequency.
- The conduction control circuit determines the output pattern for each of the upper and lower sides of the U, V and W phases.
- The protection control circuit controls emergency output stop by EMG input and OVV input.
- The dead time control circuit prevents a short circuit which may occur when the upper side and lower side are switched.
- The sync trigger generation circuit generates sync trigger signals to the AD converter.

17.3.1.1   PMD Enable Register (PMD1MDEN)

(0x4005 0480)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	PWMEN
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R/W
After reset	0	0	0	0	0	0	0	0

<PWMEN>: Enables or disables waveform synthesis.
    0: Disable
    1: Enable

Output ports that are used for the PMD become high impedance when the PMD is disabled.
Before enabling the PMD, Setting <PWMEN>="1"(enable) other relevant settings, such as output port polarity.

17.3.1.2   Port Output Mode Register (PMD1PORTMD)

(0x4005 0484)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	PORTMD	
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R/W	
After reset	0	0	0	0	0	0	0	0

<PORTMD>: Port control setting
    00: Upper phases = High-Z, lower phases = High-Z
    01: Upper phases = High-Z, lower phases = PMD output
    10: Upper phases = PMD output, lower phases = High-Z
    11: Upper phases = PMD output, lower phases = PMD output

    The <PORTMD> setting controls external port outputs of the upper phases (U, V and W phases) and the lower phases (X, Y and Z phases). When a tool break occurs while "High-Z" is selected, the upper and lower phases of external output ports are set to high impedance. In other cases, external port outputs depend on PMD outputs.
    * When PWMEN=0, output ports are set to high impedance regardless of the output port setting.
    * When an EMG input occurs, external port outputs are controlled depending on the EMGMD setting.

17.3.1.3   Mode Select Register (PMD1MODESEL)

(0x4005 04A4)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	MDSEL
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R/W
After reset	0	0	0	0	0	0	0	0

<MDSEL>: Mode select register
    0: Bus mode
    1: VE mode

   This bit selects whether to load the second buffer of each double-buffered register with the register value set via the bus (bus mode) or the value supplied from the Vector Engine (VE mode). The PWM compare registers (CMPU, CMPV, CMPW), trigger compare registers (TRGCMP0, TRGCMP1) and MDOUT register are double-buffered, and the second buffers are loaded in synchronization with the PMD's internal update timing.

17.3.2 Pulse Width Modulation Circuit
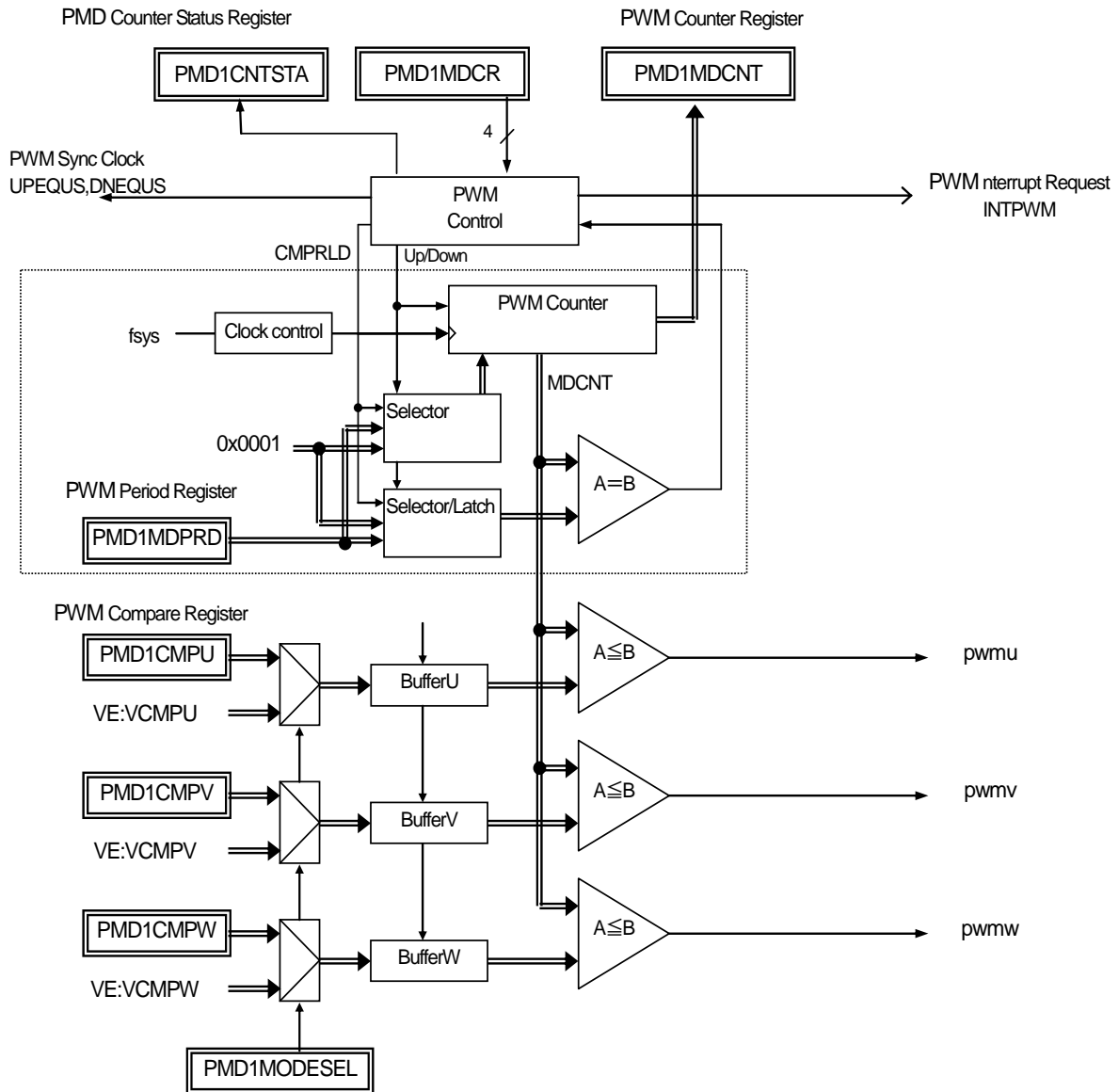
Figure 17-3 Pulse Width Modulation Circuit

The pulse width modulation circuit has a 16-bit PMD up-/down-counter and generates PWM carrier waveforms with a resolution of 12.5 nsec at 80 MHz. The PWM carrier waveform mode can be selected from mode 0 (edge-aligned PWM, sawtooth wave modulation) and mode 1 (center-aligned PWM, triangular wave modulation).

The PWM period extension mode (MDCR<PWMCK> = 1) is also available. When this mode is selected, the PWM counter generates PWM carrier waveforms with a resolution of 50 nsec.

(1) Setting the PWM period

The PWM period is determined by the MDPRD register. This register is double-buffered. Comparator input is updated at every PWM period. It is also possible to update comparator input at every half PWM period.

$$\text{Sawtooth wave PWM: MDPRD register value} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]}}$$

$$\text{Triangular wave PWM: MDPRD register value} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]} \times 2}$$

(2) Compare function

The pulse width modulation circuit compares the PWM compare registers of the 3 phases (PMD1CMPU/V/W) and the carrier wave generated by the PWM counter (PMD1MDCNT) to determine which is larger to generate PWM waveforms with the desired duty.

The PWM compare register of each phase has a double-buffered compare register. The PWM compare register value is loaded at every PWM period (when the internal counter value matches the MDPRD value). It is also possible to update the compare register at every 0.5 PWM period.
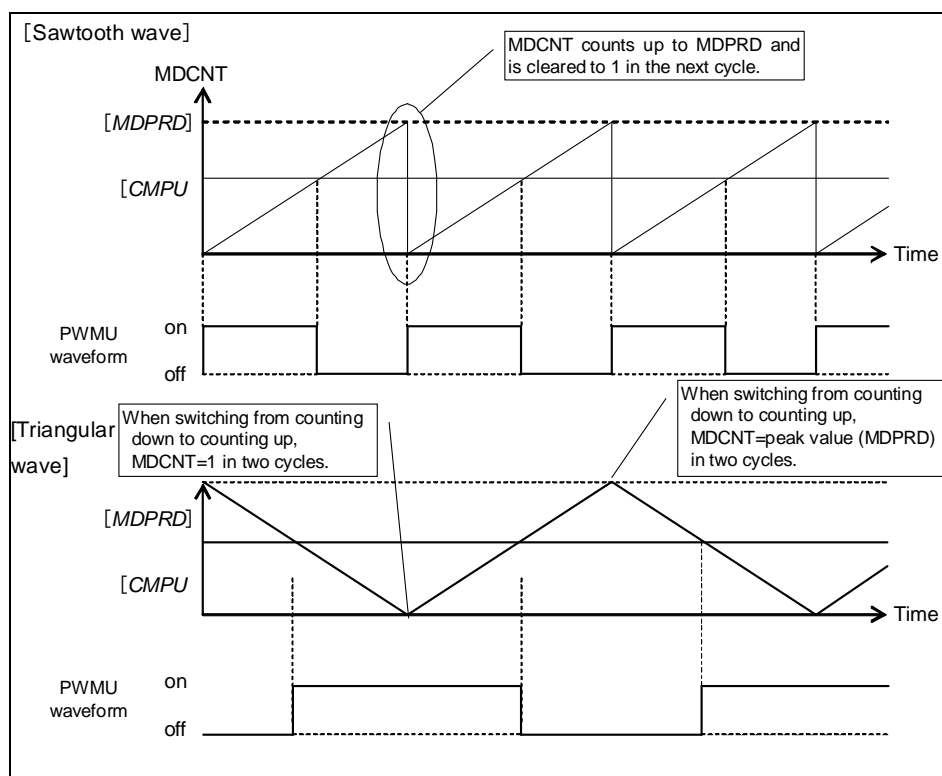
Figure 17-4 PWM Waveforms

(3) Waveform mode
Three-phase PWM waveforms can be generated in the following two modes:

i) 3-phase independent mode:
Each of the PWM compare registers for the three phases is set independently to generate independent PWM waveforms for each phase. This mode is used to generate drive waveforms such as sinusoidal waves.

ii) 3-phase common mode:
Only the U-phase PWM compare register is set to generate identical PWM waveforms for all the three phases. This mode is used for rectangular wave drive of brushless DC motors.

(4) Interrupt processing
The pulse width modulation circuit generates PWM interrupt requests in synchronization with PWM waveforms. The PWM interrupt period can be set to half a PWM period, one PWM period, two PWM periods or four PWM periods.

17.3.2.1   PMD Control Register (PMD1MDCR)

(0x4005 0488)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
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
Read/Write	R→0	R/W						
After reset	0	0	0	0	0	0	0	0

<PWMMD>: PWM carrier waveform

    0: PWM mode 0 (edge-aligned PWM, sawtooth wave)
    1: PWM mode 1(center-aligned PWM, triangular wave)
   This bit selects the PWM mode. PWM mode 0 is edge-aligned PWM and PWM mode 1 is center-aligned PWM.

<INTPRD>: PWM interrupt period

    00: Interrupt request at every 0.5 PWM period (PWM mode 1 only)
    01: Interrupt request at every PWM period
    10: Interrupt request at every 2 PWM periods
    11: Interrupt request at every 4 PWM periods
   This field selects the PWM interrupt period from 0.5 PWM period, one PWM period, two PWM periods and four PWM periods.
   When <INTPRD>=00, the contents of the compare registers (CMPU, CMPV, CMPW) and period register (MDPRD) are updated into their respective buffers when the internal counter equals 1 or the MDPRD value.

<PINT>: PWM interrupt timing

    0: Interrupt request when PWM counter = 1
    1: Interrupt request when PWM counter = MDPRD
   This bit selects whether to generate an interrupt request when the PWM counter equals its minimum or maximum value. When the edge-aligned PWM mode is selected, an interrupt request is generated when the PWM counter equals the MDPRD value. When the PWM interrupt period is set to every 0.5 PWM period, an interrupt request is generated when the PWM counter equals 1 or MDPRD.)

<DTYMD>: Duty mode

    0: 3-phase common mode
    1: 3-phase independent mode
   This bit selects whether to make duty setting independently for each phase or to use the CMPU register for all three phases.

<SYNTMD>: Port output mode
   This bit specifies the port output setting of the U, V and W phases. (See Table 17-4.)

<PWMCK>: PWM period extension mode

    0: Normal period
    1: 4x period
   When <PWMCK>=0, the PWM counter operates with a resolution of 12.5 ns at fsys=80 MHz.
            * Sawtooth wave: 12.5 ns, triangular wave: 25 ns
   When <PWMCK>=1, the PWM counter operates with a resolution of 50 ns at fsys=80 MHz.
            * Sawtooth wave: 50 ns, triangular wave: 100 ns

17.3.2.2   PWM Counter Status Register (PMD1CNTSTA)

(0x4005 048C)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	UPDWN
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R
After reset	0	0	0	0	0	0	0	0

<UPDWN>: PWM counter flag
   0: Up-counting
   1: Down-counting
   This bit indicates whether the PWM counter is up-counting or down-counting.
   When the edge-aligned PWM mode is selected, this bit is always read as 0.

17.3.2.3   PWM Counter Register (PMD1MDCNT)

(0x4005 0490)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	MDCNT							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	MDCNT							
Read/Write	R							
After reset	0	0	0	0	0	0	0	1

<MDCNT>: PWM counter
   PMD counter value (resolution: 12.5 ns at fsys = 80 MHz)
           * Sawtooth wave: 12.5 ns, triangular wave: 25 ns
           * When MDCR<PWMCK>=1, the counter resolution becomes 50 ns.
   A16-bit counter for reading the PWM period count value. It is read-only.

   * When the PMD is disabled (PWMEN=0), the value of PWM counter depends on the setting of
   PWMMD (PWM carrier waveform). The value is as follows.
           In case of PWMMD=0: 0x0001
           In case of PWMMD=1: the value of MDPRD

17.3.2.4   PWM Period Register (PMD1MDPRD)

(0x4005 0494)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	MDPRD							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	MDPRD							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<MDPRD>: PWM period

MDPRD≧0x0010

A 16-bit register for specifying the PWM period. This register is double-buffered and can be changed even when the PWM counter is operating. The buffer is loaded at every PWM period. (That is, when the PWM counter matches the MDPRD value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or MDPRD. The least significant bit must be set as 0.)

If <MDPRD> is set to a value less than 0x0010, it is automatically assumed to be 0x0010. (The register retains the actual value that is written.)

* Do not write to this register in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

17.3.2.5   PWM Compare Registers (PMD1CMPU, PMD1CMPV, PMD1CMPW)

(0x4005 0498-049B)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	CMPU							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	CMPU							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(0x4005 049C-049F)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	CMPV							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	CMPV							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(0x4005 04A0-04A3)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	CMPW							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	CMPW							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<CMPU, CMPV, CMPW>: PWM pulse width
Compare registers (resolution : 12.5 ns at fsys =80 MHz)
* Sawtooth wave: 12.5 ns, triangular wave: 25 ns
* When MDCR<PWMCK>=1, the counter resolution becomes 50 ns.

CMPU, CMPV and CMPW are compare registers for determining the output pulse width of the U, V and W phases. Theses registers are double-buffered. Pulse width is determined by comparing the buffer and the PWM counter to evaluate which is larger. (To be loaded when the PWM counter value matches the MDPRD value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or MDPRD.) When this register is read, the value of the first buffer (data set via the bus) is returned.
* To load the second buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting MODESEL<MDSEL> to 0.
* Do not write to these registers in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Motor control circuit

17.3.3 Conduction Control Circuit
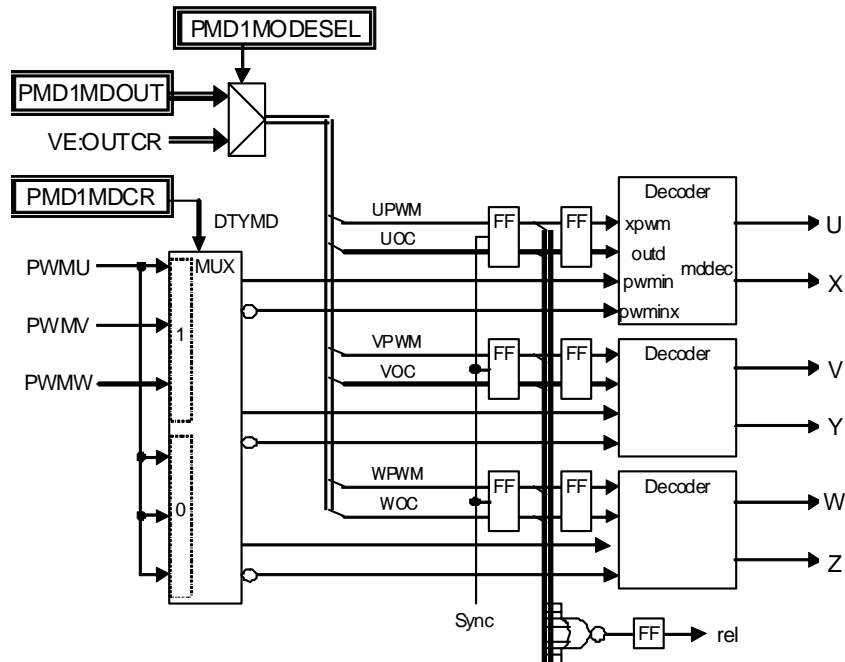
Figure 17-5 Conduction Control Circuit

The conduction control circuit performs output port control according to the settings made in the PMD output register (PMD1MDOUT). The PMD1MDOUT register bits are divided into two parts: settings for the synchronizing signal for port output and settings for port output. The latter part is double-buffered and update timing can be set as synchronous or asynchronous to PWM.

The output settings for six port lines are made independently for each of the upper and lower phases through the bits 10 to 8 of the MDOUT register and bits 3 and 2 of the MDPOT register. In addition, bits 10 to 8 of the MDOUT register select PWM or H/L output for each of the U, V and W phases. When PWM output is selected, PWM waveforms are output. When H/L output is selected, output is fixed to either a high or low level. Table 17-3 shows a summary of port outputs according to port output settings in the MDOUT register and polarity settings in the MDCR register.

17.3.3.1   PMD Output Setting Register (PMD1MDPOT)
                    (0x4005 04AC)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	POLH	POLL	PSYNCS	
Read/Write	R→0	R→0	R→0	R→0	R/W	R/W	R/W	
After reset	0	0	0	0	0	0	0	0

<PSYNCS>: MDOUT transfer timing
        00: Async to PWM
        01: Load when PWM counter = 1
        10: Load when PWM counter= MDPRD
        11: Load when PWM counter = 1 or MDPRD
    PSYNCS selects the timing when the U-, V- and W-phase output settings are reflected in port outputs
    (sync or async to the PWM counter peak, bottom or peak/bottom).
    When "00" (Async to PWM) is selected, the changing of MDOUT register is applied to the U-, V- and
    W-phase output immediately. The <PSYNCS> is also available in the vector engine.
        * This field must be set while MDEN<PWMEN>=0.
<POLL>: Lower phase port polarity
        0: Active low
        1: Active high
    POLL selects the output port polarity of the lower phases.
        * This bit must be set while MDEN<PWMEN>=0.
<POLH>: Upper phase port polarity
        0: Active low
        1: Active high
    POLH selects the output port polarity of the upper phases.
        * This bit must be set while MDEN<PWMEN>=0.

17.3.3.2   PMD Output Control Register (PMD1MDOUT)

(0x4005 04A8)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	WPWM	VPWM	UPWM
Read/Write	R→0	R→0	R→0	R→0	R→0	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	WOC		VOC		UOC	
Read/Write	R→0	R→0	R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

<UOC, VOC, WOC>,<UPWM, VPWM,WPWM>: U-, V-, and W-phase output control
   The MDOUT register controls the port outputs of the U, V and W phases (see Table 17-3 below.)
     * To load the second buffer of MDOUT with a value updated via the bus, select the bus mode (default)
       by setting MODESEL<MDSEL> to 0.
     * Do not write to this register in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written
       separately, operation cannot be guaranteed.

○PMDnMDCR<SYNTMD>=0

Polarity: Active high (MDPOT bits 3, 2=1)

MDOUT Output Control		MDOUT Bits 10, 9, 8 H/L or PWM Output Select			
Bits 5, 3, 1 Upper phase	Bits 4, 2, 0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	/PWM	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	/PWM

Polarity: Active low (MDPOT bits 3, 2=0)

MDOUT Output Control		MDOUT Bits 10, 9, 8 H/L or PWM Output Select			
Bits 5,3,1 Upper phase	Bits 4,2,0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	/PWM
0	1	H	L	H	/PWM
1	0	L	H	/PWM	H
1	1	L	L	/PWM	PWM

○PMDnMDCR<SYNTMD>=1

Polarity: Active high (MDPOT bits 3, 2=1)

MDOUT Output Control		MDOUT Bits 10, 9, 8 H/L or PWM Output Select			
Bits 5, 3, 1 Upper phase	Bits 4, 2, 0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	/PWM	PWM
0	1	L	H	L	/PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	/PWM

Polarity: Active low (MDPOT bits 3, 2=0)

MDOUT Output Control		MDOUT Bits 10, 9, 8 H/L or PWM Output Select			
Bits 5,3,1 Upper phase	Bits 4,2,0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	/PWM
0	1	H	L	H	PWM
1	0	L	H	/PWM	H
1	1	L	L	/PWM	PWM

Table 17-3 Port Outputs according to the UOC, VOC, WOC, UPWM, VPWM and WPWM Settings

Output Settings for Center-off PWM
  Center-off PWM can be supported by the following settings.

Table 17-4 Register Settings for Center-off PWM

	Normal PWM cennter on	U-Phase PWM center off	V-Phase PWM centert off	W-Phase PWM center offt
CMPU	duty_U	MDPRD-duty_U	duty_U	duty_U
CMPV	duty_V	duty_V	MDPRD-duty_V	duty_V
CMPW	duty_W	duty_W	duty_W	MDPRD-duty_W
UOC	11	00	11	11
VOC	11	11	00	11
WOC	11	11	11	00

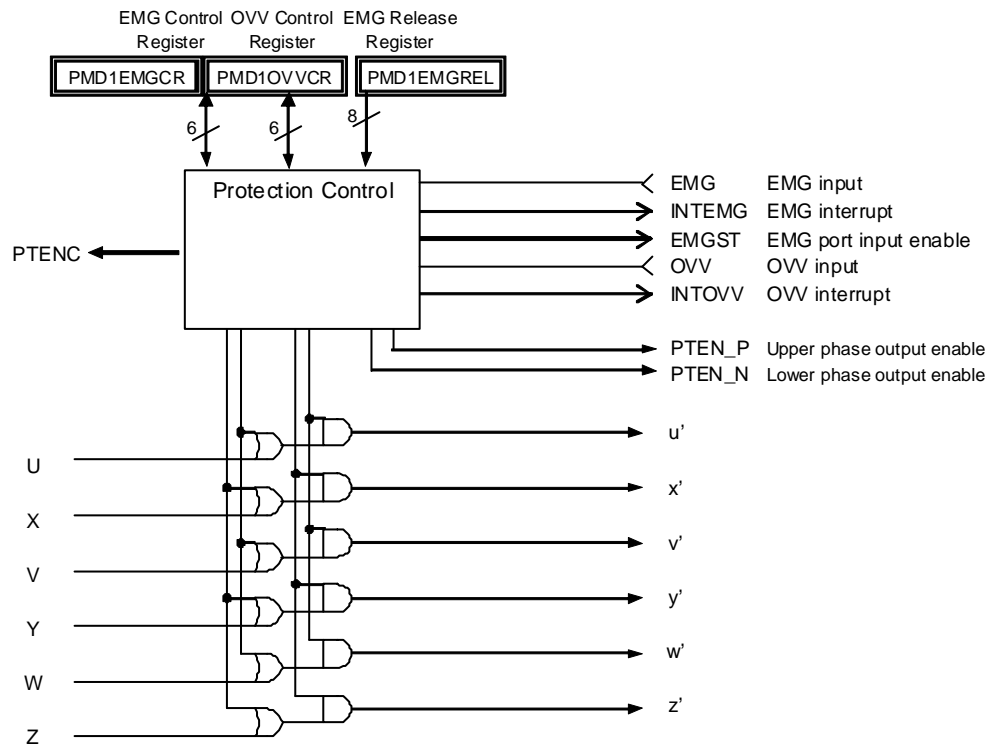## 17.3.4 Protection Control Circuit

Figure 17-6 Protection Control Circuit

The protection control circuit consists of an EMG protection control circuit and an OVV protection control circuit.

Motor control circuit

TOSHIBA   TMPM372/3/4

17.3.4.1   EMG Protection Circuit

   The EMG protection circuit consists of an EMG protection control unit and a port output disable unit. This circuit is activated when the EMG input becomes low. The EMG protection circuit offers an emergency stop mechanism: when the EMG input is asserted (H → L), all six port outputs are immediately disabled (depending on the EMGCR<EMGMD> setting) and an EMG interrupt (INTEMG) is generated. EMGCR<EMGMD> can be set to output a control signal that sets external output ports to high impedance in case of an emergency.
   A tool break also disables all six PWM output lines depending on the EMGCR<EMGMD> setting. When a tool break occurs, external output ports can be set to high impedance through the setting of the PORTMD register.
   EMG protection is set through the EMG Control Register (EMGCR). A read value of 1 in EMGSTA<EMGST> indicates that the EMG protection circuit is active. In this state, EMG protection can be released by setting all the port output lines inactive (MDOUT[10:8][5:0]) and then setting EMGCR<EMGRS> to 1.
   To disable the EMG protection function, write 0x5A and 0xA5 in this order to the EMGREL register and then clear EMGCR<EMGEN> to 0. (These three instructions must be executed consecutively.) While the EMG protection input is low, any attempt to release the EMG protection state is ignored. Before setting EMGCR<EMGRS> to 1 to release EMG protection, make sure that EMGST<EMGI> is high.
   The EMG protection circuit can be disabled only after the specified key codes (0x5A, 0xA5) are written in the EMGREL register to prevent it from being inadvertently disabled.

> **Note: Initial procedure for EMG function**
> **After reset, the EMG function is enabled but EMG pin is configured as a normal port. Therefore, as the EMG protection might be valid, release the EMG protection by the following procedure at the initial sequence.**
> **(1) Selects EMG function by PxFC register.**
>   **(2) Reads PMDnEMGSTA<EMGI> to confirm it as "1".**
>   **(3) Sets PMDnMDOUT<10:8、5:0> to "0" to make all ports in-active ("L" output).**
>   **(4) Releases EMG protection by setting PMDnEMGCR<EMGRS> to "1".**
>
> **If the EMG protection is to be disabled, continue the following procedure.**
>   **(5) Writes the key codes to PMDnEMGREL. (In order of 0x5A and 0xA5)**
>   **(6) Sets PMDnEMGCR<EMGEN> to "0" to disable the EMG protection.**

17.3.4.2   EMG Release Register (PMD1EMGREL)
                 (0x4005 04B0)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	EMGREL							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

<EMGREL>: EMG disable code
   The EMG and OVV protection functions can be disabled by setting 5A and A5 in this order to bits 7 to 0 of the EMGREL register.
   When disabling these functions, EMGCR<EMGEN> and OVVCR<OVVEN> must be cleared to 0.
     * This register is used for both the EMG and OVV functions.

TMPM372/3/4 17-20   **Motor control circuit**

17.3.4.3   EMG Control Register (PMD1EMGCR)

(0x4005 04B4)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	EMGCNT			
Read/Write	R→0	R→0	R→0	R→0	R/W			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	INHEN	EMGMD		-	EMGRS	EMGEN
Read/Write	R→0	R→0	R/W	R/W		R/W	W	R/W
After reset	0	0	1	1	1	0	0	1

<EMGEN>: EMG protection circuit enable/disable
   0: Disable
   1: Enable
   The EMG protection circuit is enabled by setting this bit to 1. In the initial state, the EMG protection
circuit is enabled.
   To disable this circuit, write 5A and A5 in this order to the EMGREL register and then clear the EMGEN
bit to 0.

<EMGRS>: EMG protection release
0: -
1: Release protection
EMG protection can be released by setting the MDOUT register to 0 and then setting the EMGRS bit to 1.
This bit is always read as 0.
* Be sure to write 0 to both the upper bits [10:8] and lower bits [5:0].
* Before releasing EMG protection, make sure that the EMG input has returned to high.

<EMGMD>: EMG protection mode select
00: PWM output control disabled / Port output = All phases High-Z
01: All upper phases ON, all lower phases OFF / Port output = Lower phases High-Z
10: All upper phases OFF, all lower phases ON / Port output = Upper phases High-Z
11: All phases OFF / Port output = All phases High-Z
* ON = PWM output (no output control), OFF = Low [when <POLL>,<POLH>=1 (active high)]
This field controls PWM output and port output of the upper and lower phases in case of an emergency.

<INHEN>: Tool break enable/disable
0: Disable
1: Enable
This bit selects whether or not to stop the PMD when the PMD stop signal is input from the tool. In the initial state, tool breaks are enabled.

<EMGCNT>: EMG input detection time
0 to 15 (When <EMGCNT=0, the noise filter is bypassed.)
EMGCNT×16/fsys (resolution: 200[nsec] at 80 MHz)

Note: Always write "0" to bit2 in PMD1EMGCR.

17.3.4.4 EMG Status Register (PMD1EMGSTA)
(0x4005 04B8)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	EMGI	EMGST
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R	R
After reset	0	0	0	0	0	0	-	0

<EMGST>: EMG protection state
0: Normal operation
1: Protected
The EMG protection state can be known by reading this bit.

<EMGI>: EMG input
EMG protection state
The EMG input state can be known by reading this bit.

17.3.4.5   OVV Protection Control Circuit (OVV)

The OVV protection control circuit consists of an OVV protection control unit and a port output disable unit. This circuit is activated when the OVV input port is asserted.

When the OVV input is asserted (H → L) for a specified period (set in OVVCR<OVVCNT>), the OVV protection circuit fixes the six port output lines in the conduction control circuit to high or low. At this time, an OVV interrupt (INTOV) is generated. It is possible to turn off only the upper or lower phases or all phases.

OVV protection is set through the OVV Control Register (OVVCR). A read value of 1 in OVVSTA<OVVST> indicates that the OVV protection circuit is active.

The release of the OVV protection state is enabled by setting OVVCR<OVVEN> to 1. Then, OVV protection is automatically released after the OVV protection circuit completes its operation.

 * The OVV protection state is not released while the OVV protection input is low. The state of this port input can be checked by reading OVVSTA<OVVI>.

The OVV protection state is released in synchronization with the PWM period (when the PWM count matches the MDPRD value). (When 0.5 PWM period is selected, the release timing is when the PWM counter equals 1 or MDPRD.)

To disable the OVV protection function, write 0x5A and 0xA5 in this order to the EMGREL register and then clear OVVCR<OVVEN> to 0. (These three instructions must be executed consecutively.)

The OVV protection circuit can be disabled only after the specified key codes (0x5A, 0xA5) are written in the EMGREL register to prevent it from being inadvertently disabled.

17.3.4.6   OVV Control Register (PMD1OVVCR)

(0x4005 04BC)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	OVVCNT			
Read/Write	R→0	R→0	R→0	R→0	R/W			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
Read/Write	R→0	R/W	R/W	R/W		R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

<OVVEN>: OVV protection circuit enable/disable
    0: Disable
    1: Enable
The OVV protection circuit is enabled by setting this bit to 1. In the initial state, the OVV protection circuit is disabled.
To disable this circuit, write 5A and A5 in this order to the EMGREL register and then clear this bit to 0. (These three instructions must be executed consecutively.)

<OVVRS>: OVV protection release

    0: Disable automatic release of OVV protection

    1: Disable automatic release of OVV protection

   The OVV protection state is entered when the overvoltage detection signal makes a high-to-low transition. After the overvoltage detection signal returns high, the OVV protection state can be automatically released by a match between the PWM counter and the MDPRD register by setting this bit to 1.

     * When 0.5 PWM period is selected (MDCR<INTPRD>=00), the OVV protection state is released when the PWM counter equals 1 or MDPRD.

<OVVISEL>: OVV input select

    0: Port input

    1: ADC monitor signal

   This bit selects whether to use port input or the monitor signal from the ADC as the OVV signal to be input to the protection circuit.

   When the ADC monitor signal is selected, <OVVCNT> becomes invalid.

<OVVMD>: OVV protection mode

    00: No output control

    01: All upper phases ON, all lower phases OFF

    10: All upper phases OFF, all lower phases ON

    11: All phases OFF

       * ON=High, OFF=Low [when <POLL>,<POLH>=1 (active high)]

   This field controls the outputs of the upper and lower phases when an OVV condition occurs.

    * If OVV and EMG conditions occur simultaneously, the protection mode settings in the EMGCR register become effective.

<ADIN0EN>: ADC A monitor interrupt input enable

    0: Disable

    1: Enable

   This bit selects whether to enable or disable the monitor signal input from ADC A.

   When this bit is set to enable and <OVVISEL>=1, the PMD is placed in a protection state (if OVV protection is enabled) by an interrupt signal from ADC A that is generated by a match between an AD conversion result and the specified compare value.

    * For details, see the chapter on the ADC.

<ADIN1EN>: ADC B monitor interrupt input enable

    0: Disable input

    1: Enable input

   This bit selects whether to enable or disable the monitor signal input from ADC B.

   When this bit is set to enable and <OVVISEL>=1, the PMD is placed in a protection state (if OVV protection is enabled) by an interrupt signal from ADC B that is generated by a match between an AD conversion result and the specified compare value.

    * For details, see the chapter on the ADC.

<OVVCNT>: OVV input detection time

   1-15 (If 0 is set, it is handled as 1.)

   OVVCNT×16/fsys (resolution: 200[nsec] at 80 MHz)

    * OVVCNT is effective only when port input is selected as the OVV signal (<OVVISEL>=1).

17.3.4.7   OVV Status Register (PMD1OVVSTA)

(0x4005 04C0)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	OVVI	OVVST
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R	R
After reset	0	0	0	0	0	0	-	0

<OVVST>: OVV protection state
  0: Normal operation
  1: Protected
  The OVV state can be known by reading this bit.
<OVVI>: OVVI input
 OVVI state
  The OVV input state (selected by OVVCR<OVVISEL>) can be known by reading this bit.
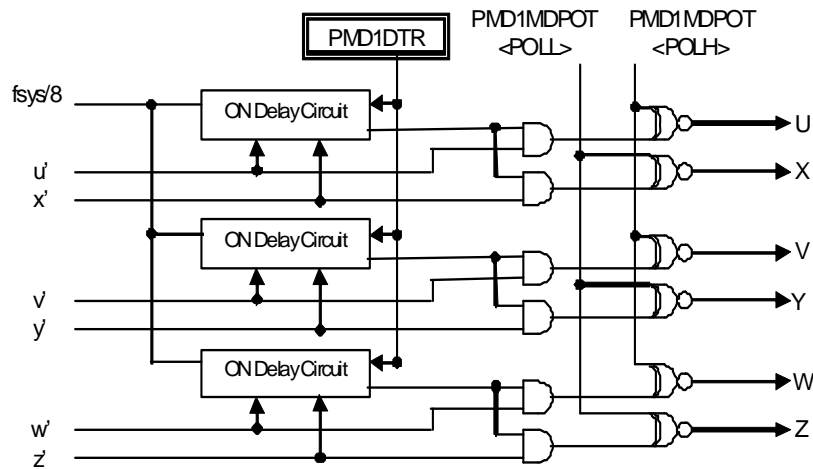
17.3.5 Dead Time Circuit

Figure 17-6 Dead Time Circuit

The dead time circuit consists of a dead time unit and an output polarity switching unit.
For each of the U, V and W phases, the ON delay circuit introduces a delay (dead time) when the upper and lower phases are switched to prevent a short circuit. The dead time is set to the Dead Time Register (DTR) as an 8-bit value with a resolution of 100 ns at 80 MHz.
The output polarity switching circuit allows the polarity (active high or active low) of the upper and lower phases to be independently set through MDPOT<POLH> and <POLL>.

17.3.5.1 Dead Time Register (PMD1DTR)
(0x4005 04C4)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	DTR							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<DTR>: Dead time
100 nsec × 8 bits (up to 25.5 μsec at fsys = 80 MHz)

* Do not change this register while MDEN<PWMEN>=1.
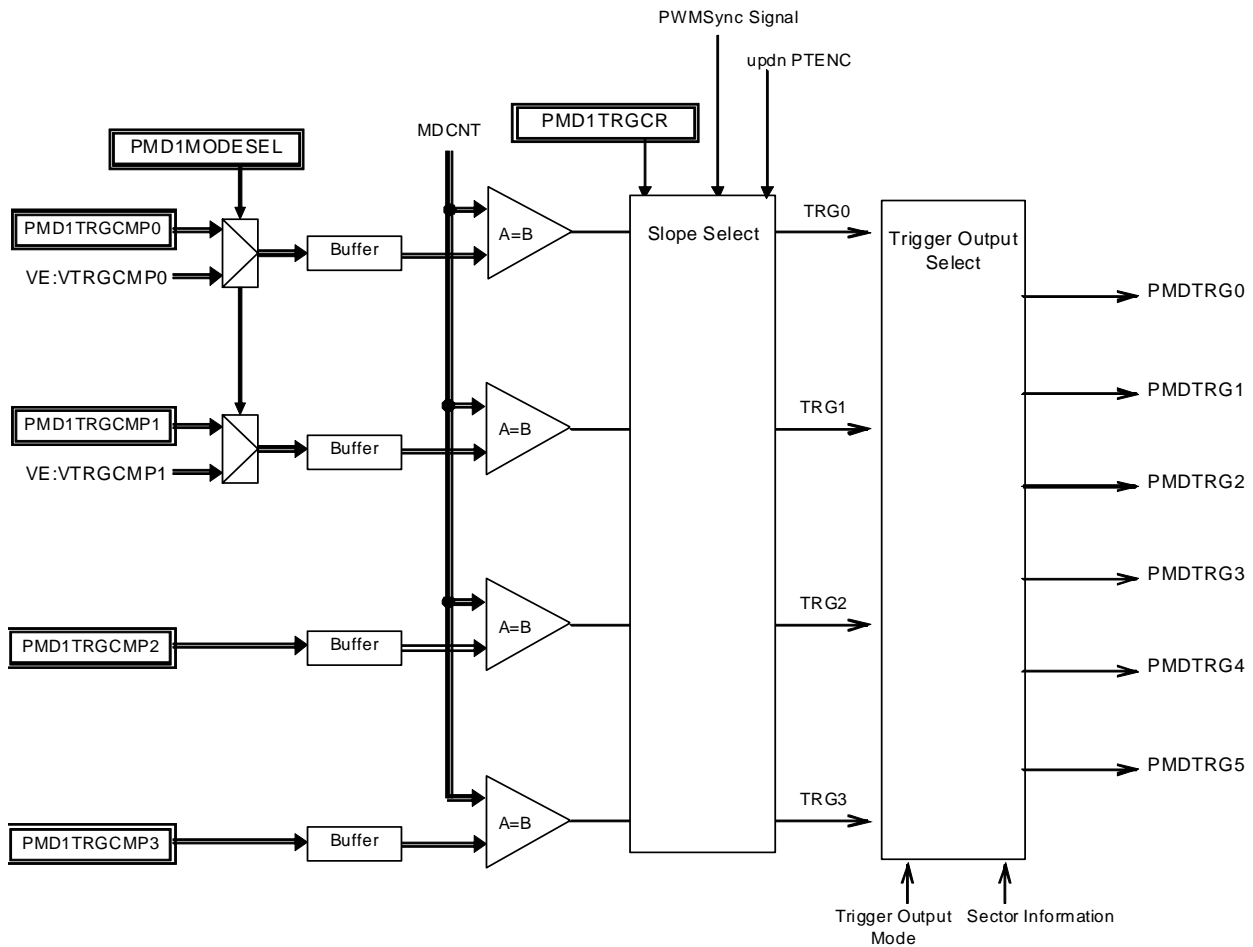
17.3.6 Sync Trigger Generation Circuit

Figure 17-7 Sync Trigger Generation Circuit

The sync trigger generation circuit generates trigger signals for starting ADC sampling in synchronization with PWM. The ADC trigger signal (PMDTRG) is generated by a match between MDCNT and TRGCMP. The signal generation timing can be selected from up-count match, down-count match and up-/down-count match. When the edge-aligned PWM mode is selected, the ADC trigger signal is generated on an up-count match. When PWM output is disabled (MDEN<PWMEN>=0), trigger output is also disabled.

When the trigger select output mode is selected, the trigger output port is switched according to the TRGSEL register setting or sector information from the Vector Engine.

17.3.6.1   Trigger Compare Registers (PMD1TRGCMP0 ~3)

(0x4005 04C8-04CB)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRGCMP0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRGCMP0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(0x4005 04CC-04CF)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRGCMP1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRGCMP1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(0x4005 04D0-04D3)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRGCMP2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRGCMP2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(0x4005 04D4-04D7)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRGCMP3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRGCMP3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<TRGCMP0-3>: Trigger output compare registers

When the PMD counter value (MDCNT) matches the value set in TRGCMPx, PMDTRG is output. When TRGCMPx is read, the value in the first buffer of the double buffers (data set via the bus) is returned.
  TRGCMPx should be set in a range of 1 to [MDPRD set value – 1].
   * It is prohibited to set TRGCMPx to 0 or the MDPRD value.
   * To load the data in TRGCMP0 and TRGCMP1 to the second buffers, select the bus mode (default) by setting MODESEL<MDSEL> to 0.
   * Do not write to these registers in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
   * When TRGCMPx is set to 0x0001, no trigger output is made only in the first cycle after PWM start (MDEN<PWMEN>1).

Update Timing of the Trigger Compare Register (TRGCMPx)
  The Trigger Compare Register (TRGCMPx) is double-buffered. The timing at which the data written to TRGCMPx is loaded to the second buffer depends on the setting of TRGCR<TRGxMD>. When TRGCR<TRGxBE> is set to 1, data written to TRGCMPx is immediately loaded to the second buffer.

Table 17-6 TRGCMPx Buffer Update Timing according to Trigger Output Mode Setting

TRGxMD	TBUFx Update Timing
000:Trigger output disabled	Always updated
001:Trigger output on down-count match	Updated when PWM counter equals MDPRD (PWM carrier peak)
010:Trigger output on up-count match	Updated when PWM counter equals 1 (PWM carrier bottom)
011:Trigger output on up-/down-count match	Updated when PWM counter equals 1 or MDPRD (PWM carrier peak/bottom)
100:Trigger output at PWM carrier peak	Always updated
101:Trigger output at PWM carrier bottom	
110:Trigger output at PWM carrier peak/bottom	
111:Trigger output disabled	

17.3.6.2　Trigger Control Register (PMD1TRGCR)

(0x4005 04D8)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
Read/Write	R/W	R/W			R/W	R/W		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
Read/Write	R/W	R/W			R/W	R/W		
After reset	0	0	0	0	0	0	0	0

<TRG0MD, TRG1MD, TRG2MD, TRG3MD>: PMDTRG0 to PMDTRG3 mode setting
000: Trigger output disabled
001: Trigger output at down-count match
010: Trigger output at up-count match
011: Trigger output at up-/down-count match
100: Trigger output at PWM carrier peak
101: Trigger output at PWM carrier bottom
110: Trigger output at PWM carrier peak/bottom
111: Trigger output disabled

This register selects trigger output timing.
When the PMD is set to the edge-aligned mode, trigger outputs are made on up-count match or at PWM carrier peak even if down-count match or PWM carrier bottom is selected.

* When <TRGxMD>=011, TRGCMPx=0x0001 and MDCR<PWMMD>=1 (triangular wave), one trigger output is made per period.

When the 1-shunt mode is used, the acceptable PMDTRG is as follows.

VEFMODE1 <IDMODE>	PMD1TRGCR <TRG6MD>	PMD1TRGCR <TRG7MD>
10	010 (up-count)	010 (up-count)
10	101 (carrier bottom)	010 (up-count)
11	001 (down-count)	001 (down-count)
11	001 (down-count)	101 (carrier bottom)

< TRG0BE、TRG1BE、TRG2BE、TRG3BE >: PMDTRG0 to PMDTRG3 buffer update timing
0: Sync
1: Async (The value written to PMDTRGx is immediately reflected.)

This bit enables asynchronous updating of the PMDTRG0 to PMDTRG3 buffers.

17.3.6.3   Trigger Output Mode Setting Register (PMD1TRGMD)

(0x4005 04DC)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	TRGOUT	EMGTGE
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R/W	R/W
After reset	0	0	0	0	0	0	0	0

<EMGTGE>: Output enable in EMG protection state
    0: Disable trigger output in the protection state
    1: Enable trigger output in the protection state

    This bit enables or disables trigger output in the EMG protection state.

<TRGOUT>: Trigger output mode
    0: Fixed trigger output
    1: Variable trigger output

    When <TRGOUT>=0, trigger outputs PMDTRG0 to PMDTRG3 output the trigger signals generated by
a match with TRGCMP0 to TRGCMP3 respectively.PMDTRG4 and PMDTRG5 are fixed to a low level.
    When <TRGOUT>=1, trigger output by TRGCMP0 is switched according to the TRGSEL register
setting or sector information from the Vector Engine. For details, see the table below.

Table 17-7 Trigger Output Patterns

TRGOUT Setting	Compare Register	TRGSEL Setting	Trigger Output
TRGOUT=0	TRGCMP0	X	PMDTRG0
	TRGCMP1		PMDTRG1
	TRGCMP2		PMDTRG2
	TRGCMP3		PMDTRG3
TRGOUT=1	TRGCMP0	0	PMDTRG0
		1	PMDTRG1
		2	PMDTRG2
		3	PMDTRG3
		4	PMDTRG4
		5	PMDTRG5
	TRGCMP1	x	No trigger output
	TRGCMP2	x	No trigger output
	TRGCMP3	x	No trigger output

Motor control circuit

17.3.6.4   Trigger Output Select Register (PMD1TRGSEL)

(0x4005 04E0)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	TRGSEL		
Read/Write	R→0	R→0	R→0	R→0	R→0	R/W		
After reset	0	0	0	0	0	0	0	0

<TRGSEL>: Trigger output select
000: Output from PMDTRG0
001: Output from PMDTRG1
010: Output from PMDTRG2
011: Output from PMDTRG3
100: Output from PMDTRG4
101: Output from PMDTRG5
110: No trigger output
111: No trigger output

This field is effective when the variable trigger output mode is selected (TRGMD<TRGOUT>=1). The selected trigger is output by a match between the PMD counter and the TRGCMP0 value. (See Table 1 7 -6.)

18. 12-Bit Analog-to-Digital Converters

The TMPM372/373/374 contains one 12-bit successive-approximation analog-to-digital converters (ADC).

The ADC unit B (ADC B) has 11 analog inputs for TMPM372, 7 analog inputs for TMPM373 and 6 analog inputs for TMPM374.

Functions and features

(1) It can select analog input and start AD conversion when receiving trigger signal from PMD or TMRB (interrupt).
(2) It can select analog input, in the Software Trigger Program and the Constant Trigger Program.
(3) The ADC has twelve register for AD conversion result.
(4) The ADC generates an interrupt signal at the end of the program which was started by PMD trigger and TMRB trigger.
(5) The ADC generates an interrupt signal at the end of the program which are the Software Trigger Program and the Constant Trigger Program.
(6) The ADC has the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

18.1 **Block Diagram**

The following shows a block diagram of the ADC.
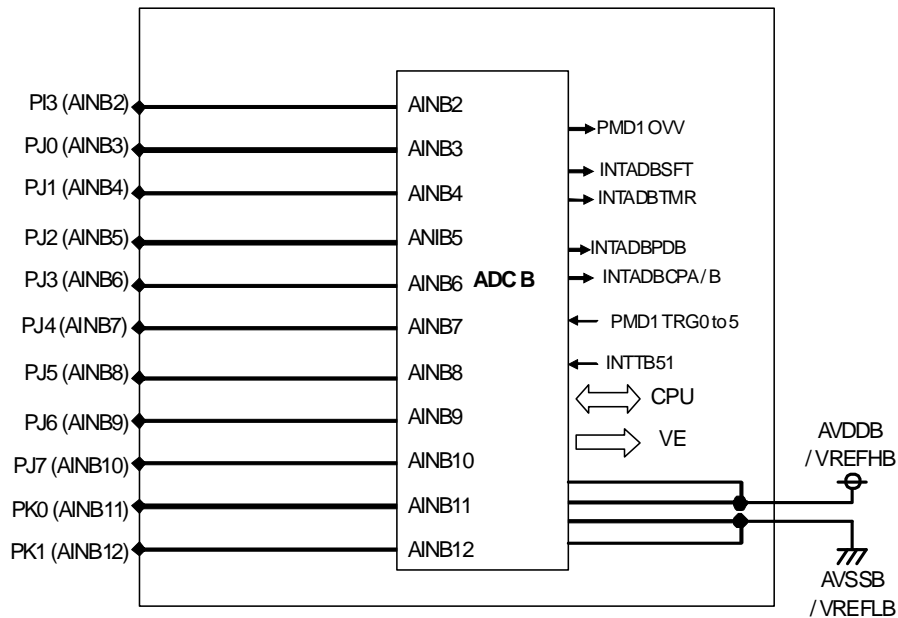
Note 1: TMPM373 doesn't have AINB4, AINB5, AINB6 and AINB7 input.
Note 2: TMPM374 doesn't have AINB4, AINB5, AINB6, AINB7 and AINB8 input.

Fig 18.1 AD converters Block Diagram

18.2 List of Registers

The ADC has the following registers.

Table 18.1 ADC Registers

| Address	Register Name	Description
4003_02_00	ADBCLK	ADC Clock Setting Register
4003_02_04	ADBMOD0	ADC Mode Setting Register 0
4003_02_08	ADBMOD1	ADC Mode Setting Register 1
4003_02_0C	ADBMOD2	ADC Mode Setting Register 2
4003_02_10	ADBCMPCR0	Monitoring Setting Register 0
4003_02_14	ADBCMPCR1	Monitoring Setting Register 1
4003_02_18	ADBCMP0	AD Conversion Result Compare Register 0
4003_02_1C	ADBCMP1	AD Conversion Result Compare Register 1
4003_02_20	ADBREG0	AD Conversion Result Register 0
4003_02_24	ADBREG1	AD Conversion Result Register 1
4003_02_28	ADBREG2	AD Conversion Result Register 2
4003_02_2C	ADBREG3	AD Conversion Result Register 3
4003_02_30	ADBREG4	AD Conversion Result Register 4
4003_02_34	ADBREG5	AD Conversion Result Register 5
4003_02_38	ADBREG6	AD Conversion Result Register 6
4003_02_3C	ADBREG7	AD Conversion Result Register 7
4003_02_40	ADBREG8	AD Conversion Result Register 8
4003_02_44	ADBREG9	AD Conversion Result Register 9
4003_02_48	ADBREG10	AD Conversion Result Register 10
4003_02_4C	ADBREG11	AD Conversion Result Register 11
4003_02_50	-	Reserved
4003_02_54	-	Reserved
4003_02_58	-	Reserved
4003_02_5C	-	Reserved
4003_02_60	-	Reserved
4003_02_64	-	Reserved
4003_02_68	ADBPSEL6	PMD Trigger Program Number Select Register 6
4003_02_6C	ADBPSEL7	PMD Trigger Program Number Select Register 7
4003_02_70	ADBPSEL8	PMD Trigger Program Number Select Register 8
4003_02_74	ADBPSEL9	PMD Trigger Program Number Select Register 0
4003_02_78	ADBPSEL10	PMD Trigger Program Number Select Register 10
4003_02_7C	ADBPSEL11	PMD Trigger Program Number Select Register 11
4003_02_80	ADBPINTS0	PMD Trigger Interrupt Select Register 0
4003_02_84	ADBPINTS1	PMD Trigger Interrupt Select Register 1
4003_02_88	ADBPINTS2	PMD Trigger Interrupt Select Register 2
4003_02_8C	ADBPINTS3	PMD Trigger Interrupt Select Register 3
4003_02_90	ADBPINTS4	PMD Trigger Interrupt Select Register 4
4003_02_94	ADBPINTS5	PMD Trigger Interrupt Select Register 5
4003_02_98	ADBPSET0_REG	PMD Trigger Program Register 0
4003_02_9C	ADBPSET1_REG	PMD Trigger Program Register 1
4003_02_A0	ADBPSET2_REG	PMD Trigger Program Register 2
4003_02_A4	ADBPSET3_REG	PMD Trigger Program Register 3
4003_02_A8	ADBPSET4_REG	PMD Trigger Program Register 4
4003_02_AC	ADBPSET5_REG	PMD Trigger Program Register 5
4003_02_B0	ADBTSET_REG03	Timer Trigger Program Registers 0 to 3
4003_02_B4	ADBTSET_REG47	Timer Trigger Program Registers 4 to 7
4003_02_B8	ADBTSET_REG811	Timer Trigger Program Registers 8 to 11
4003_02_BC	ADBSSET_REG03	Software Trigger Program Registers 0 to 3
4003_02_C0	ADBSSET_REG47	Software Trigger Program Registers 4 to 7
4003_02_C4	ADBSSET_REG811	Software Trigger Program Registers 8 to 11
4003_02_C8	ADBASET_REG03	Constant Conversion Program Registers 0 to 3
4003_02_CC	ADBASET_REG47	Constant Conversion Program Registers 4 to 7
4003_02_D0	ADBASET_REG811	Constant Conversion Program Registers 8 to 11
4003_02_D4	-	Reserved

Note: Do not access to "Reserved" address.

18.3 Register Descriptions

AD conversion is performed at the clock frequency selected in the ADC Clock Setting Register.

18.3.1   ADC Clock Setting Register (ADBCLK)

ADBCLK 0x4003_0200		7	6	5	4	3	2	1	0
	Bit symbol	-	TSH3	TSH2	TSH1	TSH0	ADCLK2	ADCLK1	ADCLK0
	Read/Write	R	R/W				R/W		
	After reset	0	1001				000		
	Function	Always read as 0.	Write "1001"				AD prescaler output (SCLK) select 000: fc (Note1) 001: fc/2 010: fc/4 011: fc/8 1XX: fc/16		
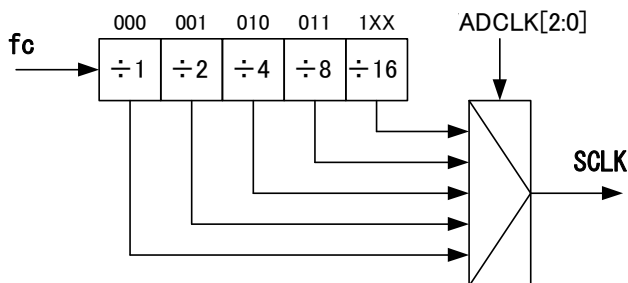
Note 1: Frequency of SCLK can be use up to 40MHz. Do not set ADCLK[2:0] to "000" when fc=80MHz.
Note 2: AD conversion is performed at the clock frequency selected in this register. The conversion clock frequency must be selected to ensure the guaranteed accuracy.
Note 3: The conversion clock must not be changed while AD conversion is in progress.

18.3.2   Mode Setting Registers

The ADC Mode Setting Registers (ADBMOD0, ADBMOD1, ADBMOD2 and ADBMOD3) are used to select how AD conversion is started.

ADBMOD0

ADBMOD0 0x4003_0204		7	6	5	4	3	2	1	0
	Bit symbol	-						DACON	ADSS
	Read/Write	R						R/W	W
	After reset	0						0	0
	Function	Always read as 0.						DAC control 0: off 1: On	Software triggered conversion 0: Don't care 1: Start

Setting <DACON> to "1", when using the ADC.
Setting <ADSS> to "1" starts AD conversion (software triggered conversion). Receiving trigger signal from PMD or TMRB(interrupt) starts AD conversion also.
For detail setting, please read the chapter about PMD and TMRB.

ADBMOD1

ADBMOD1 0xc4003_0208		7	6	5	4	3	2	1	0
	Bit symbol	ADEN	-						ADAS
	Read/Write	R/W	R						R/W
	After reset	0	0						0
	Function	AD conversion control 0: Disable 1: Enable	Always read as 0.						Constant 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.

ADBMOD2

ADBMOD2 0x4003_020C		7	6	5	4	3	2	1	0
	Bit symbol	-						ADSFN	ADBFN
	Read/Write	R						R	R
	After reset	0						0	0
	Function	Always read as 0.						Software conversion busy flag 0: Conversion completed 1: Conversion in progress	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress

The <ADBFN> is an AD conversion busy flag. When AD conversion is started regardless of conversion factor (PMD, Timer, Software, Constant), <ADBFN> is set to "1". When finished AD conversion, <ADBFN> is cleared to "0".

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".

18.3.3   Monitoring Setting Registers

The ADC has the AD conversion result monitoring function.

18.3.3.1 **ADBCMPCR0, ADBCMPCR1**

The ADBCMPCR0 and ADBCMPCR1 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.
After fixing the conversion result, the interrupt signal (INTADBCPA, INTADBCPB) is generated.

ADBCMPCR0

ADBCMPCR0
0x4003_0210

	7	6	5	4	3	2	1	0
Bit symbol	CMP0EN	-	-	ADBIG0	REGS03	REGS02	REGS01	REGS00
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Monitoring function 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Comparison condition 0: Larger than or equal to compare register 1: Smaller than or equal to compare register	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	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	CMPCNT03	CMPCNT02	CMPCNT01	CMPCNT00
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Comparison count for determining the result 0000: After every comparison 0001: After two comparisons . . 1111: After 16 comparisons			

ADBCMPCR1

ADBCMPCR1
0x4003_0214

	7	6	5	4	3	2	1	0
Bit symbol	CMP1EN	-	-	ADBIG1	REGS13	REGS12	REGS11	REGS10
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Monitoring function 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Comparison condition 0: Larger than or equal to compare register 1: Smaller than or equal to compare register	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	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	CMPCNT13	CMPCNT12	CMPCNT11	CMPCNT10
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Comparison count for determining the result 0000: After every comparison 0001: After two comparisons . . 1111: After 16 comparisons			

18.3.3.2 Conversion Result Compare Register

The ADBCMP0 and ADBCMP1 registers specify the value to be compared with an AD conversion result. The upper 12 bits (bits 4 to 15) are used.

ADBCMP0

ADBCMP0
0x4003_0218

	7	6	5	4	3	2	1	0
Bit symbol	AD0CMP03	AD0CMP02	AD0CMP01	AD0CMP00	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of the value to be compared with an AD conversion result				Always read as 0.			
	15	14	13	12	11	10	9	8
Bit symbol	AD0CMP11	AD0CMP10	AD0CMP09	AD0CMP08	AD0CMP07	AD0CMP06	AD0CMP05	AD0CMP04
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of the value to be compared with an AD conversion result							

ADBCMP1

ADBCMP1
0x4003_021C

	7	6	5	4	3	2	1	0
Bit symbol	AD1CMP03	AD1CMP02	AD1CMP01	AD1CMP00	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of the value to be compared with an AD conversion result				Always read as 0.			
	15	14	13	12	11	10	9	8
Bit symbol	AD1CMP11	AD1CMP10	AD1CMP09	AD1CMP08	AD1CMP07	AD1CMP06	AD1CMP05	AD1CMP04
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of the value to be compared with an AD conversion result							

18.3.4   AD Conversion Result Registers

18.3.4.1 ADBREG0 to ADBREG11

The ADBREGn (n = 0 to 11) register is used to store the result of an AD conversion. Bit 0 <ADRnRF> is a flag that is set when an AD conversion result is stored in the ADBREGn register and is cleared when the low-order byte of ADBREGn is read. Bit 1 <OVRn> is an overrun flag. This flag is set when a new AD conversion result is stored before the low-order byte of ADBREGn is read and is cleared when the low-order byte of ADBREGn is read.

There are twelve ADBREGn registers, which are all functionally equivalent. Only the ADBREG0 register is explained here.

AD Conversion Result Register (ADBREG0)

ADBREG0
0x4003_0220

	7	6	5	4	3	2	1	0
Bit symbol	ADR003	ADR002	ADR001	ADR000	-	-	OVR0	ADR0RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR011	ADR010	ADR009	ADR008	ADR007	ADR006	ADR005	ADR004
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG1)

ADBREG1
0x4003_0224

	7	6	5	4	3	2	1	0
Bit symbol	ADR103	ADR102	ADR101	ADR100	-	-	OVR1	ADR1RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR111	ADR110	ADR109	ADR108	ADR107	ADR106	ADR105	ADR104
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG2)

ADBREG2
0x4003_0228

	7	6	5	4	3	2	1	0
Bit symbol	ADR203	ADR202	ADR201	ADR200	-	-	OVR2	ADR2RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR211	ADR210	ADR209	ADR208	ADR207	ADR206	ADR205	ADR204
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG3)

ADBREG3
0x4003_022C

	7	6	5	4	3	2	1	0
Bit symbol	ADR303	ADR302	ADR301	ADR300	-	-	OVR3	ADR3RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR311	ADR310	ADR309	ADR308	ADR307	ADR306	ADR305	ADR304
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG4)

ADBREG4
0x4003_0230

	7	6	5	4	3	2	1	0
Bit symbol	ADR403	ADR402	ADR401	ADR400	-	-	OVR4	ADR0R4
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR411	ADR410	ADR409	ADR408	ADR407	ADR406	ADR405	ADR404
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG5)

ADBREG5
0x4003_0234

	7	6	5	4	3	2	1	0
Bit symbol	ADR503	ADR502	ADR501	ADR500	-	-	OVR5	ADR5RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR511	ADR510	ADR509	ADR508	ADR507	ADR506	ADR505	ADR504
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG6)

ADBREG6
0x4003_0238

	7	6	5	4	3	2	1	0
Bit symbol	ADR603	ADR602	ADR601	ADR600	-	-	OVR6	ADR6RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR611	ADR610	ADR609	ADR608	ADR607	ADR606	ADR605	ADR604
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG7)

ADBREG7
0x4003_023C

	7	6	5	4	3	2	1	0
Bit symbol	ADR703	ADR702	ADR701	ADR700	-	-	OVR7	ADR7RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR711	ADR710	ADR709	ADR708	ADR707	ADR706	ADR705	ADR704
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG8)

ADBREG8
0x4003_0240

	7	6	5	4	3	2	1	0
Bit symbol	ADR803	ADR802	ADR801	ADR800	-	-	OVR8	ADR8RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR811	ADR810	ADR809	ADR808	ADR807	ADR806	ADR805	ADR804
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG9)

ADBREG9
0x4003_0244

	7	6	5	4	3	2	1	0
Bit symbol	ADR903	ADR902	ADR901	ADR900	-	-	OVR9	ADR9RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR911	ADR910	ADR909	ADR908	ADR907	ADR906	ADR905	ADR904
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG10)

ADBREG10
0x4003_0248

	7	6	5	4	3	2	1	0
Bit symbol	ADR1003	ADR1002	ADR1001	ADR1000	-	-	OVR10	ADR10RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR1011	ADR1010	ADR1009	ADR1008	ADR1007	ADR1006	ADR1005	ADR1004
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

AD Conversion Result Register (ADBREG11)

ADBREG11
0x4003_024C

		7	6	5	4	3	2	1	0
	Bit symbol	ADR1103	ADR1102	ADR1101	ADR1100	-	-	OVR11	ADR11RF
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
	Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
		15	14	13	12	11	10	9	8
	Bit symbol	ADR1111	ADR1110	ADR1109	ADR1108	ADR1107	ADR1106	ADR1105	ADR1104
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
	Function	Bits 4 to 11 of an AD conversion result							

18.3.5   PMD Trigger Program Registers

AD conversion can be started by a trigger from the PMD (programmable motor driver).

The PMD trigger program registers are used to specify the program to be started by six triggers generated by the PMD1, 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 (ADBPSELp), PMD Trigger Interrupt Select Register (ADBPINTSn) and PMD Trigger Program Setting Register (ADBPSETnm).   (p=6 to 11, n=0 to 5, m=0 to 3)

PMD Trigger Program Registers

						PMD Trigger Program Number Select Registers
ADBPSEL6	ADBPSEL7	ADBPSEL8	ADBPSEL9	ADBPSEL10	ADBPSEL11	

Program 0	Program 1	Program 2	Program 3	Program 4	Program 5	PMD Trigger Interrupt Setting Registers
ADBPINTS0	ADBPINTS1	ADBPINTS2	ADBPINTS3	ADBPINTS4	ADBPINTS5	
						PMD Trigger Program Setting Registers 0-5
ADBPST00	ADBPST10	ADBPST20	ADBPST30	ADBPST40	ADBPST50	Conversion Result Register 0
ADBPST01	ADBPST11	ADBPST21	ADBPST31	ADBPST41	ADBPST51	Conversion Result Register 1
ADBPST02	ADBPST12	ADBPST22	ADBPST32	ADBPST42	ADBPST52	Conversion Result Register 2
ADBPST03	ADBPST13	ADBPST23	ADBPST33	ADBPST43	ADBPST53	Conversion Result Register 3

The PMD Trigger Program Number Select Register (ADBPSELn) specifies the program to be started by six AD conversion start signals corresponding to six triggers generated by the PMD1.

The PMD Trigger Interrupt Select Register (ADBPINTSn) selects the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

The PMD Trigger Program Setting Register (ADBPSETnm) specifies 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 ADBPSETn0 to ADBPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADBREG0 to ADBREG3).

18.3.5.1 PMD Trigger Program Number Select Registers

The PMD Trigger Program Number Select Registers (ADBPSEL6 to ADBPSEL11) select the program to be started (from among programs 0 to 5) by trigger inputs from PMD1.

PMD Trigger Program Number Select Register 6 (ADBPSEL6)

ADBPSEL6
0x4003_0268

	7	6	5	4	3	2	1	0
Bit symbol	PENS6					PMDS62	PMDS61	PMDS60
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 7 (ADBPSEL7)

ADBPSEL7
0x4003_026C

	7	6	5	4	3	2	1	0
Bit symbol	PENS7					PMDS72	PMDS71	PMDS70
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 7 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 8 (ADBPSEL8)

ADBPSEL8
0x4003_0270

	7	6	5	4	3	2	1	0
Bit symbol	PENS8					PMDS82	PMDS81	PMDS80
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 9 (ADBPSEL9)

ADBPSEL9
0x4003_0274

	7	6	5	4	3	2	1	0
Bit symbol	PENS9					PMDS92	PMDS91	PMDS90
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 10 (ADBPSEL10)

ADBPSEL10		7	6	5	4	3	2	1	0
0x4003_0278	Bit symbol	PENS10					PMDS102	PMDS101	PMDS100
	Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	PMD Trigger Program Select Register 10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 11 (ADBPSEL11)

ADBPSEL11		7	6	5	4	3	2	1	0
0x4003_027C	Bit symbol	PENS11					PMDS112	PMDS111	PMDS110
	Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	PMD Trigger Program Select Register 11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010: Program 2 110, 111: Reserved 011: Program 3		

18.3.5.2 PMD Trigger Interrupt Select Registers

The PMD Trigger Interrupt Select Registers (ADBPINTS0 to ADBPINTS5) select the interrupt to be generated for each of programs 0 to 5.

ADBPINTS0 (for program 0)

ADBPINTS0 0x4003_0280		7	6	5	4	3	2	1	0
	Bit symbol							INTSEL01	INTSEL00
	Read/Write	R						R/W	
	After reset	0						0	
	Function	Always read as 0.						Interrupt select 00: No interrupt output 01: Reserved 10: INTADPDB 11: No interrupt output	

ADBPINTS1 (for program 1)

ADBPINTS1 0x4003_0284		7	6	5	4	3	2	1	0
	Bit symbol							INTSEL11	INTSEL10
	Read/Write	R						R/W	
	After reset	0						0	
	Function	Always read as 0.						Interrupt select 00: No interrupt output 01: Reserved 10: INTADPDB 11: No interrupt output	

ADBPINTS2 (for program 2)

ADBPINTS2 0x4003_0288		7	6	5	4	3	2	1	0
	Bit symbol							INTSEL21	INTSEL20
	Read/Write	R						R/W	
	After reset	0						0	
	Function	Always read as 0.						Interrupt select 00: No interrupt output 01: Reserved 10: INTADPDB 11: No interrupt output	

ADBPINTS3 (for program 3)

ADBPINTS3 0x4003_028C		7	6	5	4	3	2	1	0
	Bit symbol							INTSEL31	INTSEL30
	Read/Write	R						R/W	
	After reset	0						0	
	Function	Always read as 0.						Interrupt select 00: No interrupt output 01: Reserved 10: INTADPDB 11: No interrupt output	

ADBPINTS4 (for program 4)

ADBPINTS4		7	6	5	4	3	2	1	0
0x4003_0290	Bit symbol							INTSEL41	INTSEL40
	Read/Write				R				R/W
	After reset				0				0
	Function				Always read as 0.			Interrupt select 00: No interrupt output 01: Reserved 10: INTADPDB 11: No interrupt output	

ADBPINTS5 (for program 5)

ADBPINTS5		7	6	5	4	3	2	1	0
0x4003_0294	Bit symbol							INTSEL51	INTSEL50
	Read/Write				R				R/W
	After reset				0				0
	Function				Always read as 0.			Interrupt select 00: No interrupt output 01: Reserved 10: INTADPDB 11: No interrupt output	

18.3.5.3 PMD Trigger Program Setting Registers 0 to 5

PMD Trigger Program Registers (ADBPSET0 to 5) are comprised of four registers. These four registers are used to select the AD conversion input pin to be used and select phase of the Vector Engine. The numbers of these registers correspond to those of the Conversion Result Registers.

Setting the <ENSPnm> to 1 enables the ADBPSETnm register. The <UVWISnm> bits are used to select phase-U, phase-V or phase-W. The <AINSPnm> bits are used to select the AIN pin to be used. The numbers of the PMD Trigger Program Setting Registers correspond to those of the Conversion Result Registers.   (n= 0 to 5 , m= 0 to 3)

PMD Trigger Program Register 0 (ADBPSET0)

ADBPSET0
0x4003_0298

	7	6	5	4	3	2	1	0
Bit symbol	ENSP00	UVWIS001	UVWIS000	AINSP004	AINSP003	AINSP002	AINSP001	AINSP000
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSP01	UVWIS011	UVWIS010	AINSP014	AINSP013	AINSP012	AINSP011	AINSP010
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG1 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSP02	UVWIS021	UVWIS020	AINSP024	AINSP023	AINSP022	AINSP021	AINSP020
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSP03	UVWIS031	UVWIS030	AINSP034	AINSP033	AINSP032	AINSP031	AINSP030
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG3 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

PMD Trigger Program Register 1 (ADBPSET1)

ADBPSET1
0x4003_029C

	7	6	5	4	3	2	1	0
Bit symbol	ENSP10	UVWIS101	UVWIS100	AINSP104	AINSP103	AINSP102	AINSP101	AINSP100
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSP11	UVWIS111	UVWIS110	AINSP114	AINSP113	AINSP112	AINSP111	AINSP110
Read/Write	R/W	R/W		R/W				
After reset	0	0		0 0				
Function	REG1 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSP12	UVWIS121	UVWIS120	AINSP124	AINSP123	AINSP122	AINSP121	AINSP120
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10 : V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSP13	UVWIS131	UVWIS130	AINSP134	AINSP133	AINSP132	AINSP131	AINSP130
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG3 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

PMD Trigger Program Register 2(ADBPSET2)

ADBPSET2
0x4003_02A0

	7	6	5	4	3	2	1	0
Bit symbol	ENSP20	UVWIS201	UVWIS200	AINSP204	AINSP203	AINSP202	AINSP201	AINSP200
Read/Write	R/W	R/W		R/W				
After reset	0	0		0 0				
Function	REG0 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSP21	UVWIS211	UVWIS210	AINSP214	AINSP213	AINSP212	AINSP211	AINSP210
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG1 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSP22	UVWIS221	UVWIS220	AINSP224	AINSP223	AINSP222	AINSP221	AINSP220
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSP23	UVWIS231	UVWIS230	AINSP234	AINSP233	AINSP232	AINSP231	AINSP230
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG3 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

PMD Trigger Program Register 3(ADBPSET3)

ADBPSET3
0x4003_02A4

	7	6	5	4	3	2	1	0
Bit symbol	ENSP30	UVWIS301	UVWIS300	AINSP304	AINSP303	AINSP302	AINSP301	AINSP300
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

	15	14	13	12	11	10	9	8
Bit symbol	ENSP31	UVWIS311	UVWIS310	AINSP314	AINSP313	AINSP312	AINSP311	AINSP310
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG1 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

	23	22	21	20	19	18	17	16
Bit symbol	ENSP32	UVWIS321	UVWIS320	AINSP324	AINSP323	AINSP322	AINSP321	AINSP320
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG2 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

	31	30	29	28	27	26	25	24
Bit symbol	ENSP33	UVWIS331	UVWIS330	AINSP334	AINSP333	AINSP332	AINSP331	AINSP330
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG3 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

PMD Trigger Program Register 4(ADBPSET4)

ADBPSET4
0x4003_02A8

		7	6	5	4	3	2	1	0
	B t symbol	ENSP40	UVWIS401	UVWIS400	AINSP404	AINSP403	AINSP402	AINSP401	AINSP400
	Read/Write	R/W	R/W		R/W				
	After reset	0	0		0				
	Function	REG0 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

		15	14	13	12	11	10	9	8
	Bit symbol	ENSP41	UVWIS411	UVWIS410	AINSP414	AINSP413	AINSP412	AINSP411	AINSP410
	Read/Write	R/W	R/W		R/W				
	After reset	0	0		0				
	Function	REG1 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

		23	22	21	20	19	18	17	16
	Bit symbol	ENSP42	UVWIS421	UVWIS420	AINSP424	AINSP423	AINSP422	AINSP421	AINSP4_0
	Read/Write	R/W	R/W		R/W				
	After reset	0	0		0				
	Function	REG2 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

		31	30	29	28	27	26	25	24
	Bit symbol	ENSP43	UVWIS431	UVWIS430	AINSP434	AINSP433	AINSP432	AINSP431	AINSP430
	Read/Write	R/W	R/W		R/W				
	After reset	0	0		0				
	Function	REG3 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

PMD Trigger Program Register 5(ADBPSET5)

ADBPSET5
0x4003_02AC

	7	6	5	4	3	2	1	0
Bit symbol	ENSP50	UVWIS501	UVWIS500	AINSP504	AINSP503	AINSP502	AINSP501	AINSP500
Read/Write	R/W	R/W	R	R/W				
After reset	0	0		O				
Function	REG0 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

	15	14	13	12	11	10	9	8
Bit symbol	ENSP51	UVWIS511	UVWIS510	AINSP514	AINSP513	AINSP512	AINSP511	AINSP510
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG1 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

	23	22	21	20	19	18	17	16
Bit symbol	ENSP52	UVWIS521	UVWIS520	AINSP524	AINSP523	AINSP522	AINSP521	AINSP520
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG2 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

	31	30	29	28	27	26	25	24
Bit symbol	ENSP53	UVWIS531	UVWIS530	AINSP534	AINSP533	AINSP532	AINSP531	AINSP530
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG3 enable 0: Disable 1: Enable	Phase select (for Vector Engine) 00: Not specified 01: U 10: V 11: W		AIN select 00000:Reserved 00001:Reserved 00010:AINB2 00011:AINB3 00100:AINB4 00101:AINB5 00110:AINB6		00111:AINB7 01000:AINB8 01001:AINB9 01010:AINB10 01011:AINB11 01100:AINB12 (01101 to 11111:reserved)		

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

18.3.6   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 ADBTSETm register. The <AINSTmn> 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 : INTADBTMR is generated.

$$(m= 0 \text{ to } 11, n= 0 \text{ to } 4)$$

Timer Trigger Program Register 03 (ADBTSET03)

ADBTSET03
0x4003_02B0

	7	6	5	4	3	2	1	0
Bit symbol	ENST0	-	-	AINST04	AINST03	AINST02	AINST01	AINST00
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENST1	-	-	AINST14	AINST13	AINST12	AINST11	AINST10
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENST2	-	-	AINST24	AINST23	AINST22	AINST21	AINST20
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENST3	-	-	AINST34	AINST33	AINST32	AINST31	AINST30
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

Timer Trigger Program Register 47 (ADBTSET47)

ADBTSET47
0x4003_02B4

	7	6	5	4	3	2	1	0
Bit symbol	ENST4	-	-	AINST44	AINST43	AINST42	AINST41	AINST40
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG4 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENST5	-	-	AINST54	AINST53	AINST52	AINST51	AINST50
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG5 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENST6	-	-	AINST64	AINST63	AINST62	AINST61	AINST60
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENST7	-	-	AINST74	AINST73	AINST72	AINST71	AINST70
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG7 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

Timer Trigger Program Register 811 (ADBTSET811)

ADBTSET811
0x4003_02B8

	7	6	5	4	3	2	1	0
Bit symbol	ENST8	-	-	AINST84	AINST83	AINST82	AINST81	AINST80
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENST9	-	-	AINST94	AINST93	AINST92	AINST91	AINST90
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENST10	-	-	AINST104	AINST103	AINST102	AINST101	AINST100
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENST11	-	-	AINST114	AINST113	AINST112	AINST111	AINST110
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

18.3.7   Software Trigger Program Registers

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. Setting the <ENSSn> to "1" enables the ADBSSETn register. The <AINSSnm> are used to select the AIN pin to be used. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers.

When finished this AD conversion, interrupt :INTADBSFT is generated.

(n= 0 to 11, m= 0 to 4)

Software Trigger Program Register 03 (ADBSSET03)

ADBSSET03
0x4003_02BC

	7	6	5	4	3	2	1	0
Bit symbol	ENSS0	-	-	AINSS04	AINSS03	AINSS02	AINSS01	AINSS00
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSS1	-	-	AINSS14	AINSS13	AINSS12	AINSS11	AINSS10
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSS2	-	-	AINSS24	AINSS23	AINSS22	AINSS21	AINSS20
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSS3	-	-	AINSS34	AINSS33	AINSS32	AINSS31	AINSS30
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

Software Trigger Program Register 47 (ADBSSET47)

ADBSSET47
0x4003_02C0

	7	6	5	4	3	2	1	0
Bit symbol	ENSS4	-	-	AINSS44	AINSS43	AINSS42	AINSS41	AINSS40
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG4 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSS5	-	-	AINSS54	AINSS53	AINSS52	AINSS51	AINSS50
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG5 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSS6	-	-	AINSS64	AINSS63	AINSS62	AINSS61	AINSS60
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSS7	-	-	AINSS74	AINSS73	AINSS72	AINSS71	AINSS70
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG7 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

Software Trigger Program Register 811 (ADBSSET811)

ADBSSET811
0x4003_02C4

	7	6	5	4	3	2	1	0
Bit symbol	ENSS8	-	-	AINSS84	AINSS83	AINSS82	AINSS81	AINSS80
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSS9	-	-	AINSS94	AINSS93	AINSS92	AINSS91	AINSS90
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSS10	-	-	AINSS104	AINSS103	AINSS102	AINSS101	AINSS100
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSS11	-	-	AINSS114	AINSS113	AINSS112	AINSS111	AINSS110
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	O				
Function	REG11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

18.3.8   Constant Conversion Program Registers

 The ADC allows conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to "1" enables the ADBASETm register. The <AINSAnm> 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, n= 0 to 4)

Constant Conversion Program Register 03 (ADBASET03)

ADBASET0
0x4003_02C8

	7	6	5	4	3	2	1	0
Bit symbol	ENSA0	-	-	AINSA04	AINSA03	AINSA02	AINSA01	AINSA00
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	7	6	5	4	3	2	1	0
Bit symbol	ENSA1	-	-	AINSA14	AINSA13	AINSA12	AINSA11	AINSA10
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	7	6	5	4	3	2	1	0
Bit symbol	ENSA2	-	-	AINSA24	AINSA23	AINSA22	AINSA21	AINSA20
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	7	6	5	4	3	2	1	0
Bit symbol	ENSA3	-	-	AINSA34	AINSA33	AINSA32	AINSA31	AINSA30
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

Constant Conversion Program Register 47 (ADBASET47)

ADBASET47
0x4003_02CC

	7	6	5	4	3	2	1	0
Bit symbol	ENSA4	-	-	AINSA44	AINSA43	AINSA42	AINSA41	AINSA40
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG4 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	7	6	5	4	3	2	1	0
Bit symbol	ENSA5	-	-	AINSA54	AINSA53	AINSA52	AINSA51	AINSA50
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG5 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	7	6	5	4	3	2	1	0
Bit symbol	ENSA6	-	-	AINSA64	AINSA63	AINSA62	AINSA61	AINSA60
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	7	6	5	4	3	2	1	0
Bit symbol	ENSA7	-	-	AINSA74	AINSA73	AINSA72	AINSA71	AINSA70
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG7 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

Constant Conversion Program Register 811 (ADBASET811)

ADBASET8
0x4003_02D0

	31	30	29	28	27	26	25	24
Bit symbol	ENSA8	-	-	AINSA84	AINSA83	AINSA82	AINSA81	AINSA80
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSA9	-	-	AINSA94	AINSA93	AINSA92	AINSA91	AINSA90
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSA10	-	-	AINSA104	AINSA103	AINSA102	AINSA101	AINSA100
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSA11	-	-	AINSA114	AINSA113	AINSA112	AINSA111	AINSA110
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000:Reserved 00111:AINB7 00001:Reserved 01000:AINB8 00010:AINB2 01001:AINB9 00011:AINB3 01010:AINB10 00100:AINB4 01011:AINB11 00101:AINB5 01100:AINB12 00110:AINB6 (01101 to 11111:reserved)				

Note 1: Do not select AINB4 to AINB7 in TMPM373.
Note 2: Do not select AINB4 to AINB8 in TMPM374.

18.4 Operation Descriptions

18.4.1   Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the AVDD5B and AVSSB pins are used. There are no registers for controlling current between AVDD5B and AVSSB. Inputs to these pins are fixed.

(Note 1) During AD conversion, do not change the output data of port I/J/K, to avoid the influence on the conversion result.

(Note 2) AD conversion results might be unstable by the following conditions.

Input operation is executed.

Output operation is executed.

Output current of port varies.

Take a countermeasure such as averaging the multiple conversion results, to get precise value.

18.4.2   Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

・PMD trigger (See "18.3.5 PMD Trigger Program Setting Registers.")

・ Timer trigger (TMRB5) (See "18.3.6 Timer Trigger Program Setting Registers.")

・Software trigger (See "18.3.7 Software Trigger Program Registers.")

These start triggers are given priorities as shown below.

PMD trigger 0 > • • • > PMD trigger 5 > Timer trigger > Software trigger > constant trigger

If the PMD trigger occurs while an AD conversion is in progress, the PMD trigger is handled stop the ongoing program and start AD conversion correspond to PMD trigger number.
If a higher-priority trigger occurs while an AD conversion is in progress, the higher-priority 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.
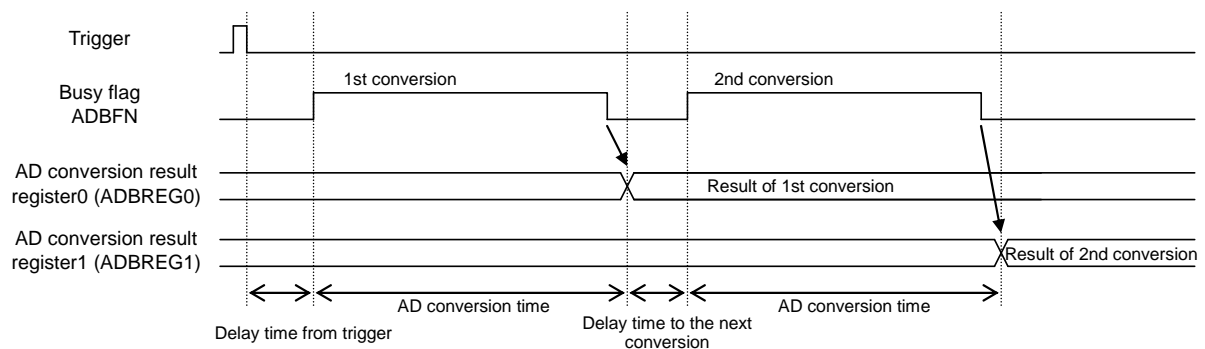
Figure Timing chart of A/D conversion

Table A/D conversion time (SCLK=40MHz)

[μs]

			fsys=80MHz		fsys=40MHz	
		Trigger	MIN	MAX	MIN	MAX
Delay time from trigger (Note 1)		PMD	0.125	0.163	0.225	0.3
		TMRB	0.125	0.263	0.225	0.5
		Software, Constant	0.138	0.275	0.25	0.525
AD conversion time		—	1.85		1.85	
Delay time to the next conversion (Note 2)		PMD	0.1	0.125	0.175	0.225
		TMRB, Software, Constant	0.1	0.238	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.

18.4.3   AD Conversion Monitoring Function

The ADC has 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 ADBCMPCR0<CMP0EN> or ADBCMPCR1<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 ADBCMCR< ADBIG0>, the interrupt (INTADBCPA for ADBCMPCR0, INTADBCPB for ADBCMPCR1) 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 (ADRxRF) is 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 (OVRx) is set.

18.5 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

18.5.1   Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADBSSET03, ADBSSET47 and ADBSSET811.
If the ADBMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.

Condition
Software trigger setting : AINB2, AINB9, AINB10, AINB11, AINB12

Software trigger conversion (<ADSS>="1")

AD conversion
AINB2 conversion   AINB9 conversion   AINB10 conversion   AINB11 conversion   AINB12 conversion

Busy flag varies according to conversion.

AD conversion busy flag (ADBFN)

Software conversion busy flag (ADSFN)

INTADSFT is generated by the completition of selected AD conversion at the timing of clearing of ADSFN.

Software trigger AD conversion interrupt (INTADBSFT)

Fig 18-3 Software trigger AD conversion

Condition
Software trigger setting : AINB10, AINB11, AINB12

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)
AINB10 conversion   AINB11 conversion   Clearing of ADBFN is delayed.

Software conversion busy flag (ADSFN)
The conversion of AINB12 doesn't start.

The result of AINB10 is stored to the result register.

The result of AINB11 is not stored to the result register.

AD conversion result register
Result register for AINB10

Fig 18-4 Writing "0" to ADEN during the software trigger AD conversion

18.5.2 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 the conversion result.

Condition
Constant conversion setting : AINB10
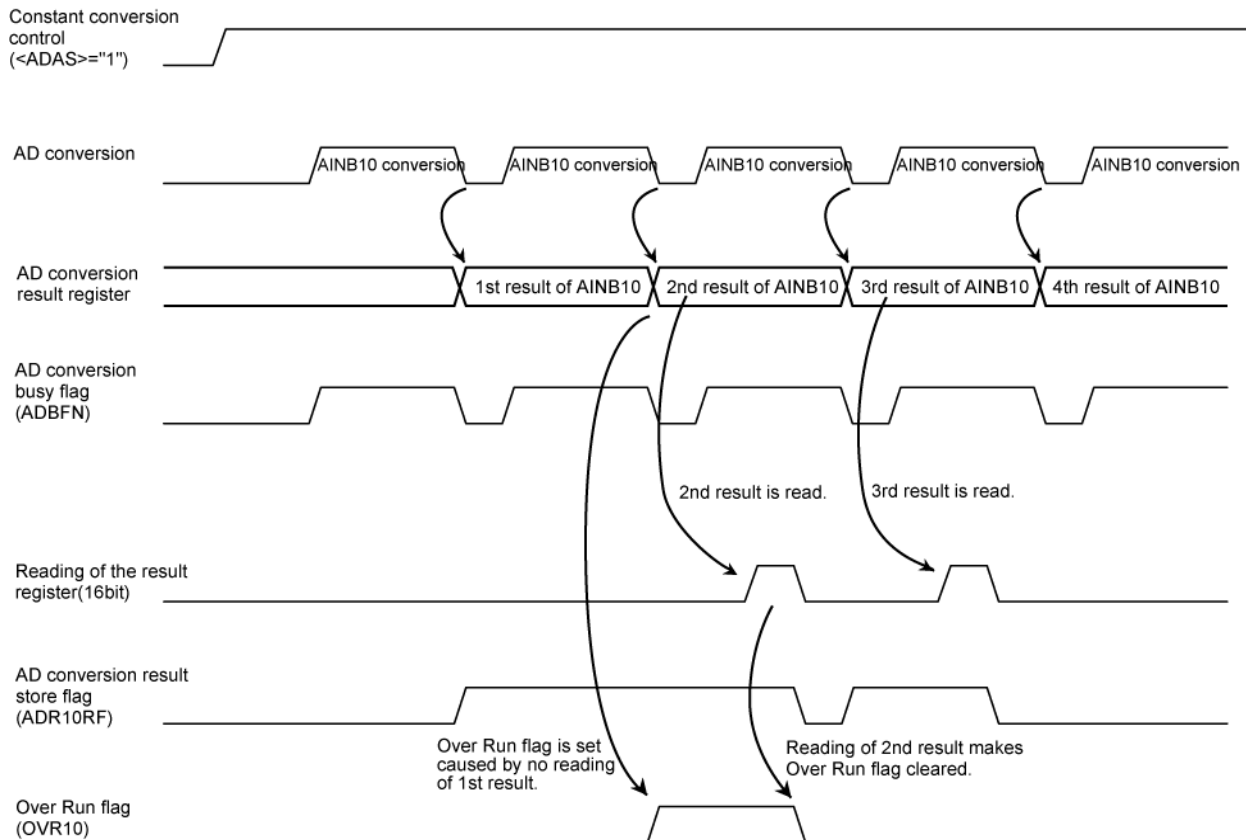
Fig 18-5 Constant conversion

18.5.3   AD conversion by trigger

If the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately.

If the timer trigger is occurred during the software trigger conversion, the ongoing conversion stops after the completion of ongoing conversion. After the completion of conversion by trigger, the software trigger conversion starts from the beginning programmed by ADxSSET03, ADxxSSET47 and ADxxSSET811.

(x=A , B : ADC unit)
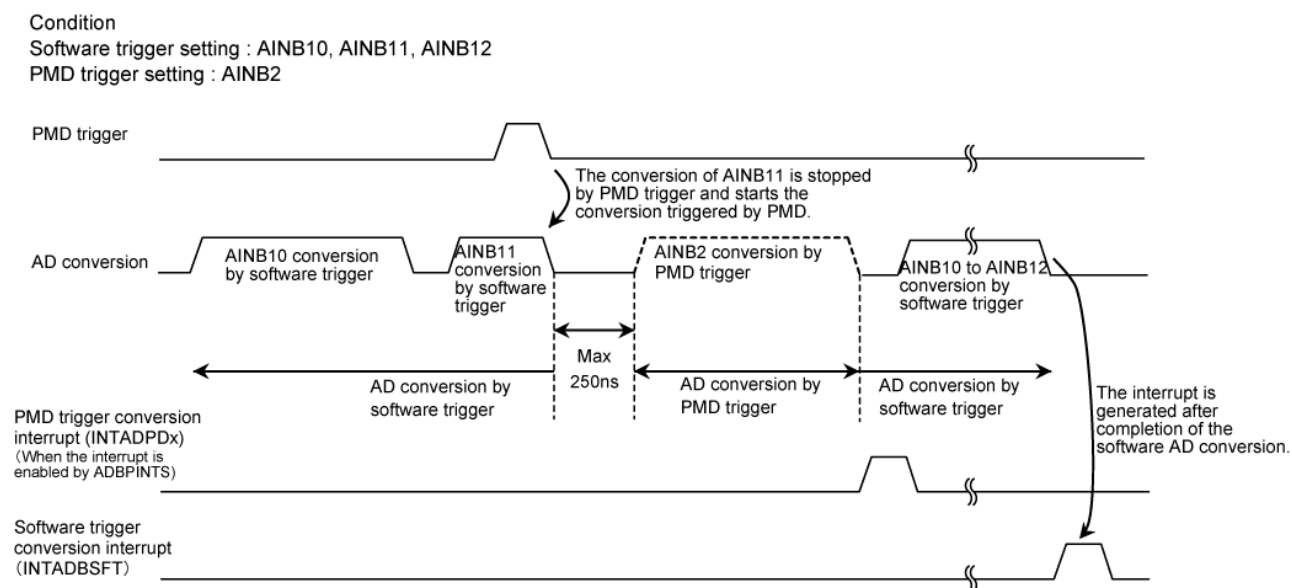
Fig 18-6 AD conversion by PMD trigger
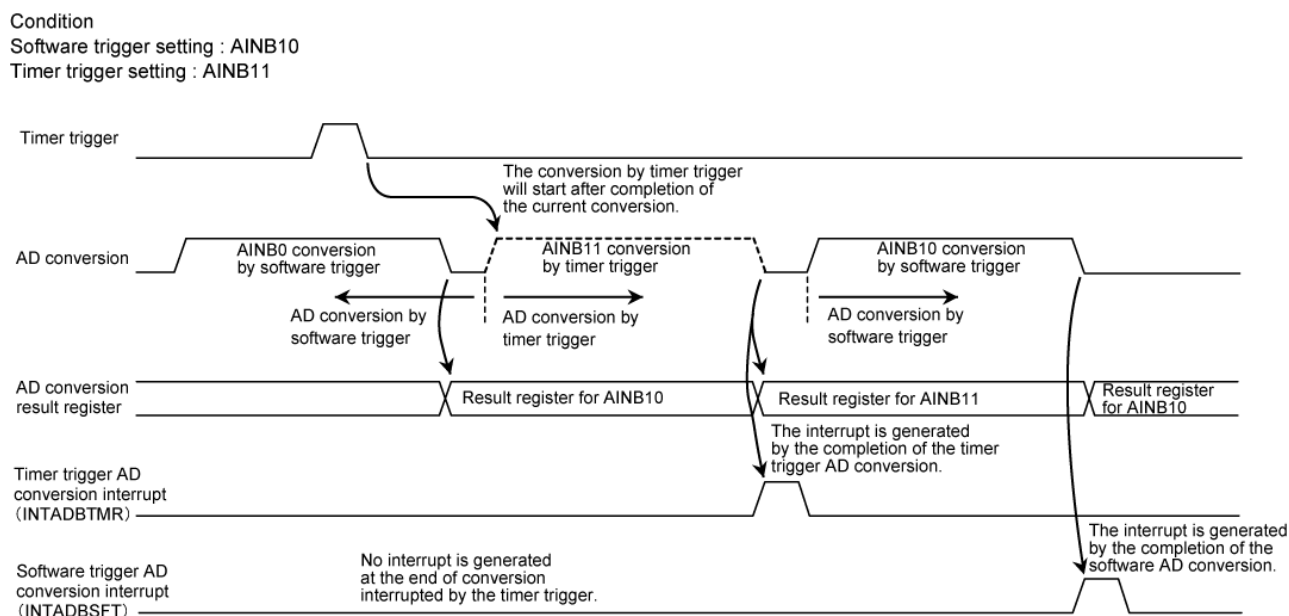
Fig 18-7 AD conversion by timer trigger (1)

Condition
Software trigger setting : AINB10, AINB11, AINB12
Timer trigger setting : AINB2
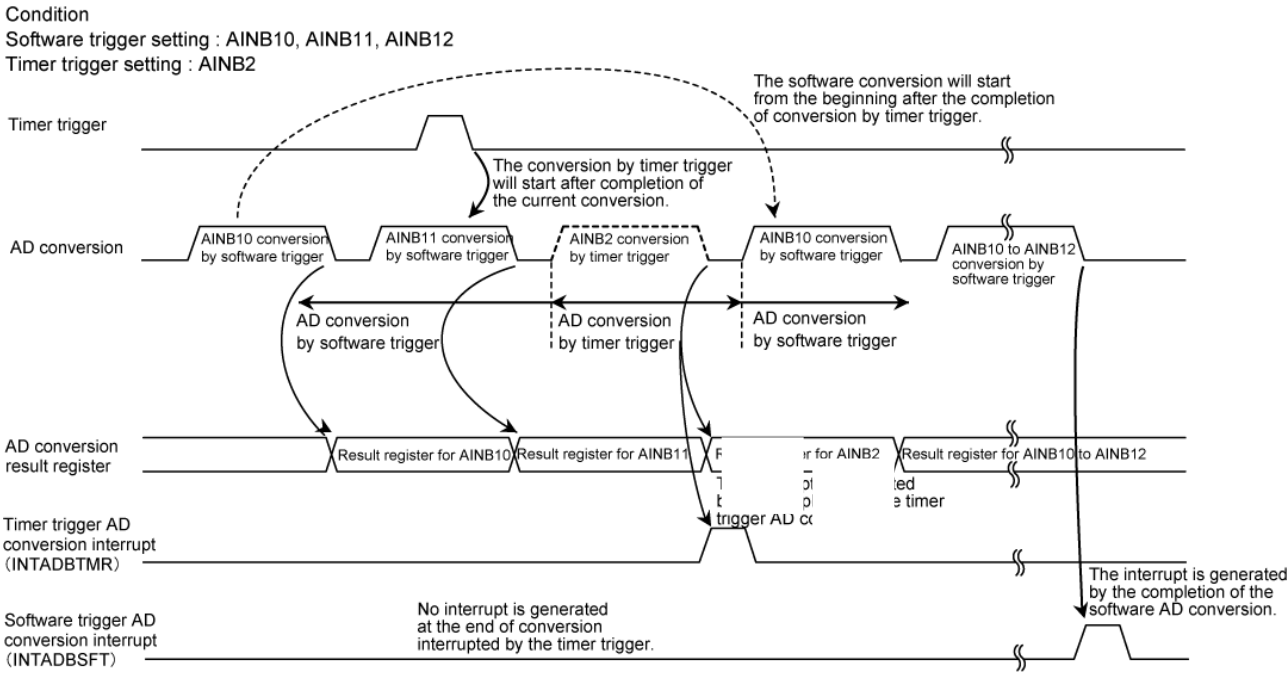
Fig 18-8 AD conversion by timer trigger (2)

19 Encoder Input Circuit (ENC)

The TMPM372/3/4FW has an incremental encoder interface, which can determine the direction and the absolute position of a motor, based on input signals from an incremental encoder.

Table 19-1 List of the ENC Registers

Register	ENC1	
	Register Symbol	Address
Encoder Input Control Register	EN1TNCR	0x4001_0500
Encoder Counter Reload Register	EN1RELOAD	0x4001_0504
Encoder Compare Register	EN1INT	0x4001_0508
Encoder Counter	EN1CNT	0x4001_050C

Table 19-2 Interrupt Sources

Interrupt Source	ENC1
ENC interrupt	INTENC1

Table 19-3 Pin Names

Pin	ENC1
Channel A input pin	PF2/ENCA1
Channel B input pin	PF3/ENCB1
Channel Z input pin	PF4/ENCZ1

19.1 Outline

The ENC can be configured to operate in one of four different modes: Encoder mode, two Sensor modes (Event count mode, Timer count mode) and Timer mode. And it also has some functions as below.

- 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) logic
- Direction discriminator
- 24-bit counter
- Comparator enable/disable
- Interrupt request output
- Digital noise filters for input signals
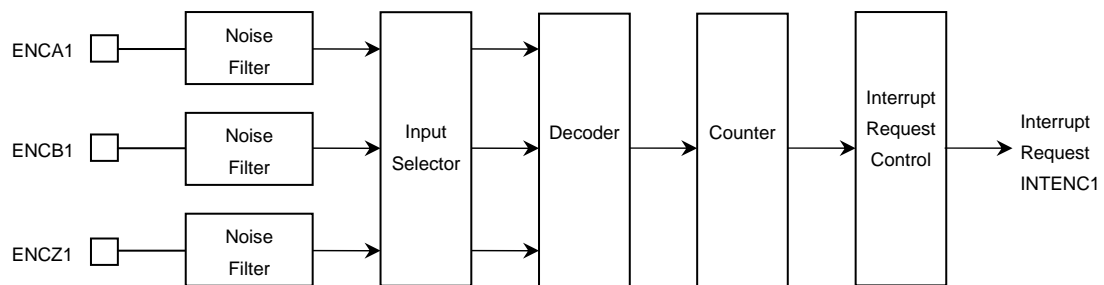
Fig 19-1   ENC Block Diagram

19.1.1 Encoder Mode

In Encoder mode, the ENC provides high-speed position tracking, based on the A/B or A/B/Z input signals from an incremental encoder.

- Event (rotation pulse) sensing: Programmable to generate an interrupt on each event.
- Event counter: Programmable to generate an interrupt at a preset count (for positional displacement calculation).
- Direction discrimination
- Up/down counting (dynamically selectable)
- Programmable counter period

19.1.2 Sensor Modes

In Sensor modes, the ENC provides low-speed position (zero-cross) tracking, based on either the U/V or U/V/W input signals from a Hall sensor.

There are two operating modes: Event Count mode and Timer Count mode (which runs with fsys).

19.1.2.1 Event Count Mode

- Event (rotation pulse) sensing: Programmable to generate an interrupt on each event.
- Event counter: Programmable to generate an interrupt at a preset count (for positional displacement calculation).
- Direction discrimination

19.1.2.2 Timer Count Mode

- Event (rotation pulse) sensing: Programmable to generate an interrupt on each event.
- Timer counting operation
- Direction discrimination
- Input capture functions

    Event capture (event interval measurement): Programmable to generate an interrupt.

    Software capture
- Event timeout error (timer compare): Programmable to generate an interrupt at a preset count.
- Revolution error: Error flag that indicates a change of the rotation direction

19.1.3 Timer Mode

The ENC can serve as a 24-bit general-purpose timer.

- 24-bit up-counter
- Counter clear: via a software clear bit, or at a preset count, or by an external trigger input, or on overflow of the free-running counter.
- Timer compare: Programmable to generate an interrupt at a preset count.
- Input capture functions

    External trigger capture: Programmable to generate an interrupt.

    Software capture

19.2 Control Registers

Encoder 1 Input Control Register

EN1TNCR		31	30	29	28	27	26	25	24
(0x4001_0500)	Bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R	R	R	R	R	R	R
	Default	0	0	0	0	0	0	0	0
	Description	Reading these bits returns a 0.							
		23	22	21	20	19	18	17	16
	Bit Symbol	-	-	-	-	-	MODE1	MODE0	P3EN
	Read/Write	R	R	R	R	R	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	Reading these bits returns a 0.					ENC Operating Mode 00: Encoder mode 01: Sensor Event Count mode 10: Sensor Timer Count mode 11: Timer mode		[In Sensor mode] 2/3-Phase Input Select 0: 2-phase 1: 3-phase
		15	14	13	12	11	10	9	8
	Bit Symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
	Read/Write	R	R	R	R	W	W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	Compare Flag 0: ― 1: Counter compared This bit is cleared on a read.	[In Sensor Timer Count mode] Revolution Error 0: ― 1: Error occurred. This bit is cleared on a read.	Rotation Direction 0: CCW 1: CW	Z_ Detected 0: Not detected 1: Z phase detected	[In Sensor Timer Count and Timer modes] Software Capture 0: ― 1: Software capture	Encoder Counter Clear 0: ― 1: Clears the counter.	[In Timer mode] Z Trigger Edge Select 0: Rising edge 1: Falling edge	Compare Enable 0: Compare disabled 1: Compare enabled
		7	6	5	4	3	2	1	0
	Bit Symbol	ZEN	ENRUN	NR1	NR0	INTEN	ENDEV2	ENDEV1	ENDEV0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	Z Phase Enable 0: Disabled 1: Enabled	ENC Run 0: Disabled 1: Enabled	Noise Filter 00: No filtering 01: Filters out pulses narrower than 31/fsys (387.5 ns@80 MHz). 10: Filters out pulses narrower than 63/fsys(787.5 ns@80 MHz). 11: Filters out pulses narrower than 127/fsys(1587 ns@80 MHz).		ENC Interrupt Enable 0: Disabled 1: Enabled	Encoder Pulse Division Factor 000: ÷1 001: ÷2 010: ÷4 011: ÷8 100: ÷16 101: ÷32 110: ÷64 111: ÷128		

Description:

<MODE1:0>: ENC Operating Mode
00: Encoder mode
01: Sensor Event Count mode
10: Sensor Timer Count mode
11: Timer mode

<MODE1:0> selects an operating mode for the ENC.

Operating modes are defined by <MODE1:0>, <P3EN> and <ZEN> as shown in the following table. There are a total of eight operating modes.

<MODE1>	<MODE0>	<ZEN>	<P3EN>	Input signal Used	Operating Mode
0	0	0	0	A, B	Encoder mode
		1		A, B, Z	Encoder mode (using Z)
0	1	0	0	U, V	Sensor Event Count mode (2 phase inputs)
			1	U, V, W	Sensor Event Count mode (3 phase inputs)
1	0	0	0	U, V	Sensor Timer Count mode (2 phase inputs)
			1	U, V, W	Sensor Timer Count mode (3 phase inputs)
1	1	0	0	-	Timer mode
		1		Z	Timer mode (using Z)

<P3EN>: 2/3-Phase Input Select
    0: 2-phase
    1: 3-phase

    <P3EN> selects the number of phase input pins used.
    If <P3EN> is cleared to 0, the ENC decodes two phase inputs.
    If <P3EN> is set to 1, the ENC decodes three phase inputs.

    Note: <P3EN> must always be cleared in Encoder mode and Timer mode , irrespective of the number of phase input pins used.

<CMP>: Compare Flag
    0: −
    1: Counter compared

    <CMP> is set to 1 when the counter value has been compared to the value programmed in the EN1INT register.
    <CMP> is cleared to 0 on a read. <CMP> remains cleared when <ENRUN> = 0. Writing to <CMP> has no effect.

<REVERR>: Revolution Error
    0: −
    1: Error occurred.

    In Sensor Timer Count mode, <REVERR> is set to 1 when a change in the rotation direction has been detected. <REVERR> is cleared to 0 on a read. <REVERR> remains cleared when <ENRUN> = 0. Writing to <REVERR> has no effect.
    This bit has an effect only in Sensor Timer Count mode.

    Note: Once software has changed the operating mode of the ENC, <REVERR> must be cleared by reading it.

<UD>: Rotation Direction
    0: Counterclockwise (CCW)
    1: Clockwise (CW)

    The quadrature signals A and B identify the motor rotation direction. <UD> is set to 1 when the CW direction is indicated (signal A of the incremental encoder signal is ahead of signal B). <UD> is cleared to 0 when the CCW direction is indicated (signal A is behind signal B). <UD> remains cleared while <ENRUN> = 0.

<ZDET>: Z Detected
        0: Not detected
        1: 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> remains cleared while <ENRUN> = 0.
<ZEN> has no influence on the value of <ZDET>.
<ZDET> remains cleared in Sensor Event Count and Sensor Timer Count modes.

<SFTCAP>: Software Capture
        0: ─
        1: Software capture

        If <SFTCAP> is set to 1, the value of the encoder counter is captured into the EN1CNT register.
Writing a 0 to <SFTCAP> has no effect. Reading <SFTCAP> always returns a 0.
In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; a write of a 1 to this bit is
ignored.

<ENCLR>: Encoder Counter Clear
        0: ─
        1: Clears the encoder counter.

        Writing a 1 to <ENCLR> clears the encoder counter to 0. Once cleared, the encoder counter restarts
counting from 0. Writing a 0 to <ENCLR> has no effect. Reading <ENCLR> always returns a 0.

<ZESEL>: Z Trigger Edge Select
        0: Uses a rising edge of the ENCZ as an external trigger input.
        1: Uses a falling edge of the ENCZ as an external trigger input.

        <ZESEL> selects the edge of the ENCZ that should be used as an external trigger in Timer mode.
In the other operating modes, <ZESEL> has no effect.

<CMPEN>: Compare Enable
        0: Compare disabled
        1: Compare enabled

        If <CMPEN> is set to 1, the value of the encoder counter is compared to the value programmed in the
EN1INT register. If <CMPEN> is cleared to 0, this comparison is not done.

<ZEN>: Z Phase Enable
        0: Disabled
        1: Enabled

        ·   In Encoder mode
          <ZEN> controls whether to clear the encoder counter (EN1CNT) on the rising or falling edge of Z.
          When <ZEN> = 1, the encoder counter is cleared on the rising edge of the ENCZ input if the
motor is rotating in the CW direction; the encoder counter is cleared on the falling edge of ENCZ if
the motor is rotating in the CCW direction. 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).

・ In Timer mode

<ZEN> controls whether to use the ENCZ signal as an external trigger input.

When <ZEN> = 1, the value of the encoder counter is captured into the EN1INT register and cleared to 0 on the edge of ENCZ selected by <ZESEL>.

In the other operating modes, <ZEN> has no effect.

<ENRUN>: ENC Run

0: Disabled

1: Enabled

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 and are not cleared even if the <ENRUN> bit is cleared to 0.

The following table shows the states of the counters and flags, depending on the value of <ENRUN>.

Internal Counter / Flag	When <ENRUN> = 0 (After reset)	When <ENRUN> = 1 (During active operation)	When <ENRUN> = 0 (During idle mode)	How to clear a counter or flag when <ENRUN> = 0
Encoder counter	0x000000	Counting	Keeps the current value.	Software clear (Write a 1 to <ENCLR>.)
Noise filter counter	0y0000000	Counting up	Counting up (Continues with noise filtering.)	Cleared only by reset.
Encoder pulse division counter	0x00	Counting down	Stopped and cleared	Cleared when <ENRUN> = 0.
Compare flag <CMP>	0	Set to 1 upon comparison; cleared to 0 on a read.	Cleared	Cleared when <ENRUN> = 0.
Revolution Error flag <REVERR>	0	Set to 1 upon an error; cleared to 0 on a read.	Cleared	Cleared when <ENRUN> = 0.
Z Detected flag <ZDET>	0	Set to 1 on detection of Z.	Cleared	Cleared when <ENRUN> = 0.
Rotation Direction Bit <UD>	0	Set or cleared according to rotation direction.	Cleared	Cleared when <ENRUN> = 0.

<NR1:0>: Noise Filter

00: No filtering

01: Filters out pulses narrower than 31/fsys as noises.

10: Filters out pulses narrower than 63/fsys as noises.

11: Filters out pulses narrower than 127/fsys as noises.

The digital noise filters remove pulses narrower than the width selected by <NR1:0>.

<INTEN>: ENC Interrupt Enable

0: Disabled

1: Enabled

<INTEN> enables and disables the ENC interrupt.

Setting <INTEN> to 1 enables interrupt generation. Clearing <INTEN> to 0 disables interrupt generation.

<ENDEV2:0>: Encoder Pulse Division Factor

    000: ÷1
    001: ÷2
    010: ÷4
    011: ÷8
    100: ÷16
    101: ÷32
    110: ÷64
    111: ÷128

The frequency of the encoder pulse is divided by the factor specified by <ENDEV2:0>.
The divided signal determines the interval of the event interrupt.

Encoder 1 Counter Reload Register

EN1RELOAD		31	30	29	28	27	26	25	24
(0x4001_0504)	Bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R	R	R	R	R	R	R
	Default	0	0	0	0	0	0	0	0
	Description	Reading these bits returns a 0.							
		23	22	21	20	19	18	17	16
	Bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R	R	R	R	R	R	R
	Default	0	0	0	0	0	0	0	0
	Description	Reading these bits returns a 0.							
		15	14	13	12	11	10	9	8
	Bit Symbol	RELOAD15	RELOAD14	RELOAD13	RELOAD12	RELOAD11	RELOAD10	RELOAD9	RELOAD8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	(See the description below.)							
		7	6	5	4	3	2	1	0
	Bit Symbol	RELOAD7	RELOAD6	RELOAD5	RELOAD4	RELOAD3	RELOAD2	RELOAD1	RELOAD0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	Setting the Encoder Counter Period (Multiplied by 4 (or 6)) 0x0000 thru. 0xFFFF When Z is used: Specifies the number of counter pulses per revolution. When Z is not used: Specifies the number of counter pulses per revolution minus 1.							

Description:

<RELOAD15:0>: setting the Encoder Counter Period

> Note: EN1RELOAD register should be accessed with 32-bit instructions.

- In Encoder mode
  <RELOAD15:0> 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 <RELOAD15:0> 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 <RELOAD15:0> on the next ENCLK.

  The EN1RELOAD register is only used in Encoder mode.

Encoder 1 Compare Register

EN1INT		31	30	29	28	27	26	25	24
(0x4001_0508)	Bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R	R	R	R	R	R	R
	Default	0	0	0	0	0	0	0	0
	Description	Reading these bits returns a 0.							
		23	22	21	20	19	18	17	16
	Bit Symbol	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	See the description below. <INT23:16> are used only in Sensor Timer Count mode and Timer mode							
		15	14	13	12	11	10	9	8
	Bit Symbol	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	See the description below.							
		7	6	5	4	3	2	1	0
	Bit Symbol	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0

	Description	In Encoder mode:
		  Generates an interrupt at the programmed encoder pulse count (0x0000 thru. 0xFFFF).
		In Sensor Event Count mode:
		  Generates an interrupt at the programmed encoder pulse count (0x0000 thru. 0xFFFF).
		In Sensor Timer Count mode:
		  Generates an interrupt when the counter has reached the programmed value without detecting a pulse (0x000000 thru. 0xFFFFFF).
		In Timer mode:
		  Generates an interrupt when the counter value has reached the programmed value (0x000000 thru. 0xFFFFFF).							

Description:

<INT15:0>: Counter Compare Value

> Note: EN1INT register should be accessed with 32-bit instructions.

- In Encoder mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT15:0>, provided <CMPEN> is set to 1. At this time, the event counter interrupt (INTENC0) is asserted if <INTEN> is set to 1.
  However, when <ZEN> = 1, INTENC is not asserted until <ZDET> is set to 1. In Encoder mode, <INT23:16> are not used (and are ignored even if programmed).

- In Sensor Event Count mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT15:0>, provided <CMPEN> is set to 1. At this time, the interrupt request (INTENC0) is asserted if <INTEN> is set to 1. The value of <ZEN> has no effect on this interrupt generation.
  In Sensor Event Count mode, <INT23:16> are not used (and are ignored even if programmed).

- In Sensor Timer Count mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT23:0>, provided <CMPEN> is set to 1. This indicates the absence of a pulse for an abnormally long period. At this time, the interrupt request (INTENC0) is asserted if <INTEN> is set to 1. The value of <ZEN> has no effect on this interrupt generation.

- In Timer mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT23:0>, provided <CMPEN> is set to 1. At this time, the timer compare interrupt (INTENC0) is asserted if <INTEN> is set to 1. The value of <ZEN> has no effect on this interrupt generation.

Encoder 1 Counter Register

EN1CNT		31	30	29	28	27	26	25	24
(0x4001_050C	Bit Symbol	-	-	-	-	-	-	-	-
	Read/Write	R	R	R	R	R	R	R	R
	Default	0	0	0	0	0	0	0	0
	Description	Reading these bits returns a 0.							
		23	22	21	20	19	18	17	16
	Bit Symbol	CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	See the description below. <CNT23:16> are used only in Sensor Timer Count mode and Timer mode. In Encoder mode and Sensor Event Count mode, reading these bits returns a 0.							
		15	14	13	12	11	10	9	8
	Bit Symbol	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0
	Description	See the description below.							
		7	6	5	4	3	2	1	0
	Bit Symbol	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Default	0	0	0	0	0	0	0	0

	Description	In Encoder mode:
		  Number of encoder pulses
		  0x0000 thru. 0xFFFF
		In Sensor mode:
		  Pulse detection time or the encoder counter value captured under software control
		  0x000000 thru. 0xFFFFFF
		In Timer mode:
		  Iencoder counter value captured by hardware signaling or under software control
		  0x000000 thru. 0xFFFFFF							

Description:

<CNT23:0>: Encoder Counter/Captured Value

> Note: EN1CNT register should be accessed with 32-bit instructions.

· In Encoder mode
  The value of encoder count.can be read out from <CNT15:0>.
  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 the value of <RELOAD15:0>, it wraps around to 0 on the next ENCLK.
  During CCW rotation, encoder counter counts down; when it has reached 0, it is reloaded with the value of <RELOAD15:0> on the next ENCLK.

· In Sensor Event Count mode
  The value of encoder count.can be read out from <CNT15:0>.
  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 reached0xFFFF, it wraps around to 0 on the next ENCLK.
  During CCW rotation, encoder counter counts down; when it has reached 0, it wraps around to 0xFFFF on the next ENCLK.

- In Sensor Timer Count mode

  <CNT23:0> contains the value of the encoder counter captured by either the encoder pulse (ENCLK) or software setting <SFTCAP> to 1. The captured value in <CNT23:0> is cleared to 0 on system reset. It can also be cleared by clearing the counter via 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 0xFFFFFF, it wraps around to 0 automatically.

- In Timer mode

  <CNT23:0> contains the value of the encoder counter captured by software setting <SFTCAP> to 1. When <ZEN> = 1, the value of the encoder counter is also captured into <CNT23:0> on the Z ENCZ edge selected by <ZESEL>.

  In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached 0xFFFFFF, it wraps around to 0 automatically.

19.3 Functional Description

19.3.1   Operating Modes

19.3.1.1   Encoder Mode

(1) When <ZEN> = 1 (<RELOAD> = 0x0380, <EN1INT> = 0x0002)
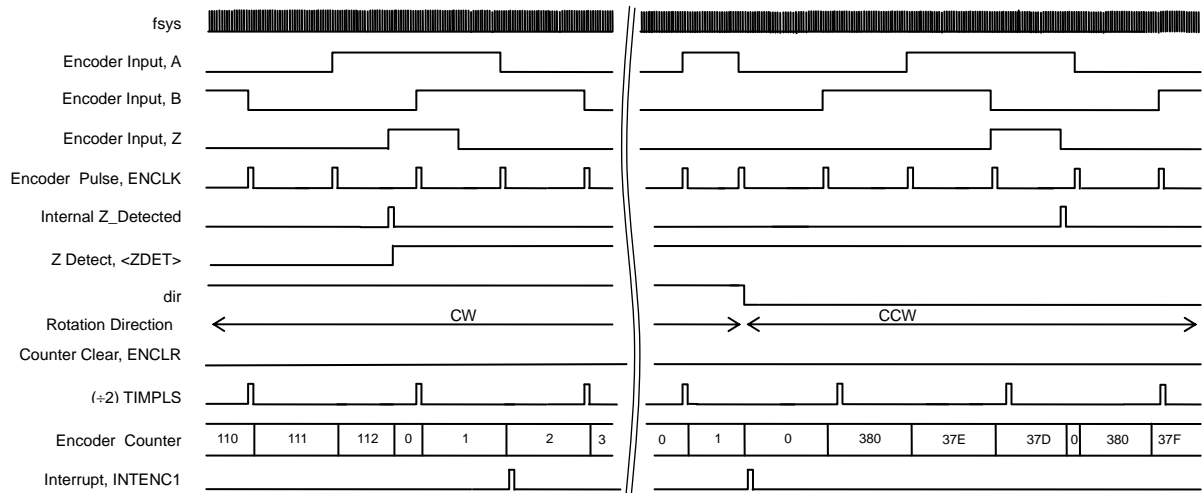
(2) When <ZEN> = 0 (<RELOAD> = 0x0380, <EN1INT> = 0x0002)
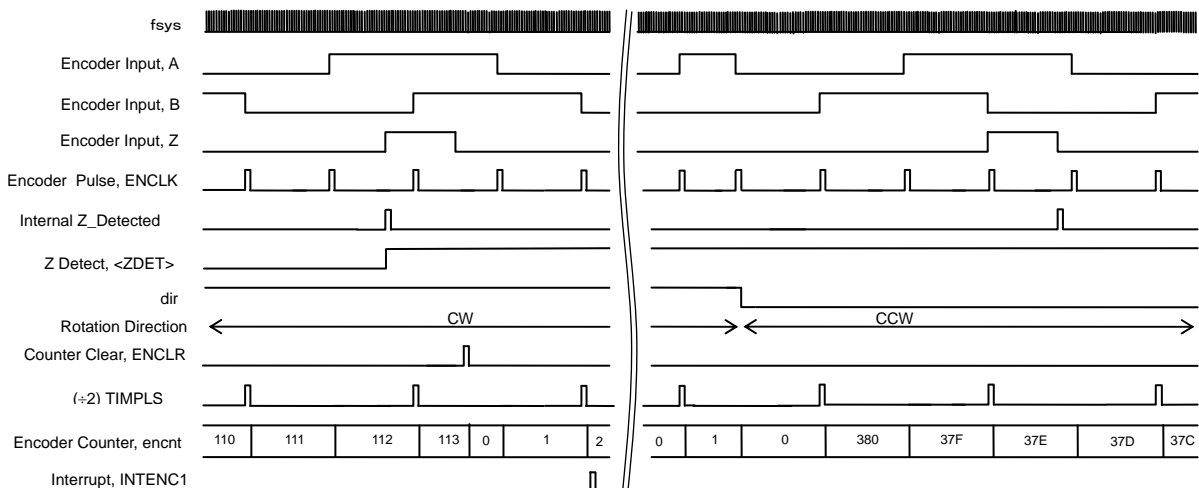
・ In Encoder mode, the incremental encoder inputs of the TMPM372/3/4 should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied_by_4 clock derived from the decoded A and B quadrature signals.

・ During CW rotation (i.e., when A leads B), the encoder counter counts up; when it has reached the value of <RELOAD>, it wraps around to 0 on the next ENCLK.

・ During CCW rotation (i.e., when A lags B), the encoder counter counts down; when it has reached 0x0000, it is reloaded with the value of <RELOAD> on the next ENCLK.

- Additionally, when <ZEN> = 1, the encoder counter is cleared to 0 on the rising edge of Z during CW rotation and on the falling edge of Z during CCW rotation (at the internal Z_Detected timing). If the ENCLK and Z edges coincide, the encoder counter is cleared to 0 without incrementing or decrementing.

- When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.

- <UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.

- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.

- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <EN1INT>. When <ZEN> = 1, however, an interrupt does not occur while <ZDET> = 0.

- Clearing <ENRUN> to 0 clears <ZDET> and <UD> to 0.

19.3.1.2 Sensor Event Count Mode

(1) When <P3EN> = 1 (<EN1INT> = 0x0002)
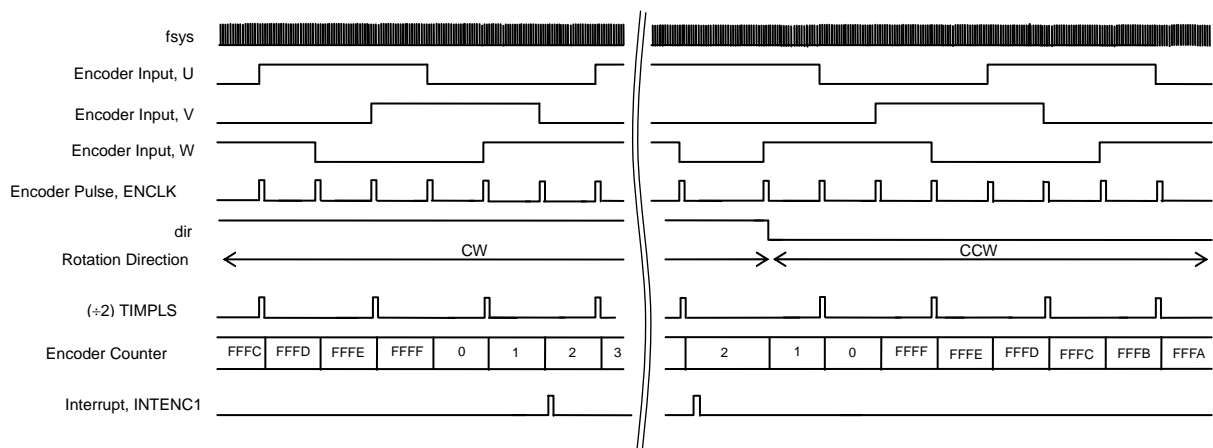
(2) When <P3EN> = 0 (<EN1INT> = 0x0002)
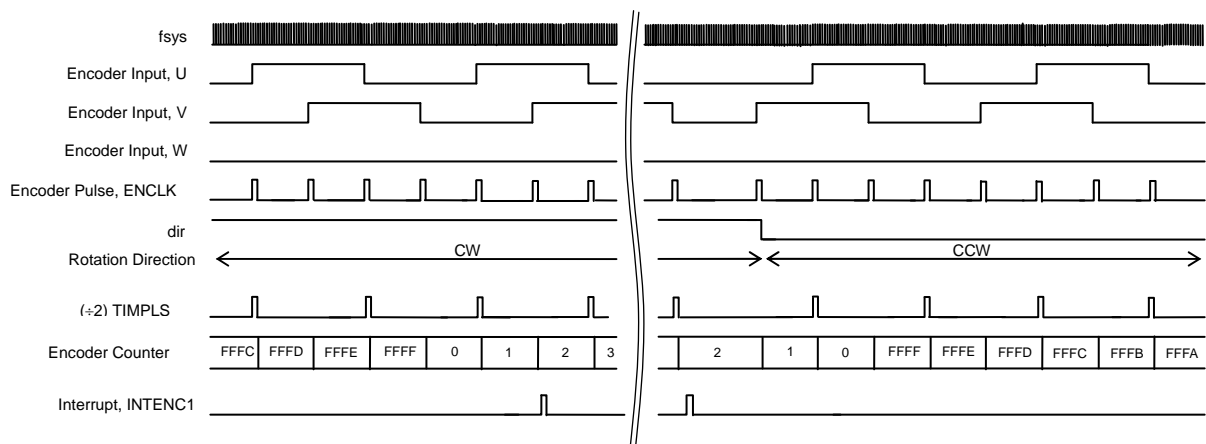
- In Sensor Event Count mode, the Hall sensor inputs of the TMPM372/3/4 should be connected to the U, V and W channels. The encoder counter counts the pulses of ENCLK, which is either multiplied_by_4 clock (when <P3EN> = 0) derived from the decoded U and V signals or multiplied_by_6 clock (when <P3EN> = 1) derived from the decoded U, V and W signals.

- During CW rotation, the encoder counter counts up; when it has reached 0xFFFF, it wraps around

to 0 on the next ENCLK.

- During CCW rotation, the encoder counter counts down; when it has reached 0x0000, it wraps around to 0xFFFF on the next ENCLK.

- When <ENCLR> is set to 1, causing the internal counter to be cleared to 0.

  - <UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.

  - TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.

  - If <CMPEN> is set to 1, an interrupt is generated when the value of the internal counter has reached the value of <EN1INT>.

- Clearing <ENRUN> to 0 clears <UD> to 0.

19.3.1.3 Sensor Timer Count Mode

(1) When <P3EN> = 1 (<EN1INT> = 0x0002)
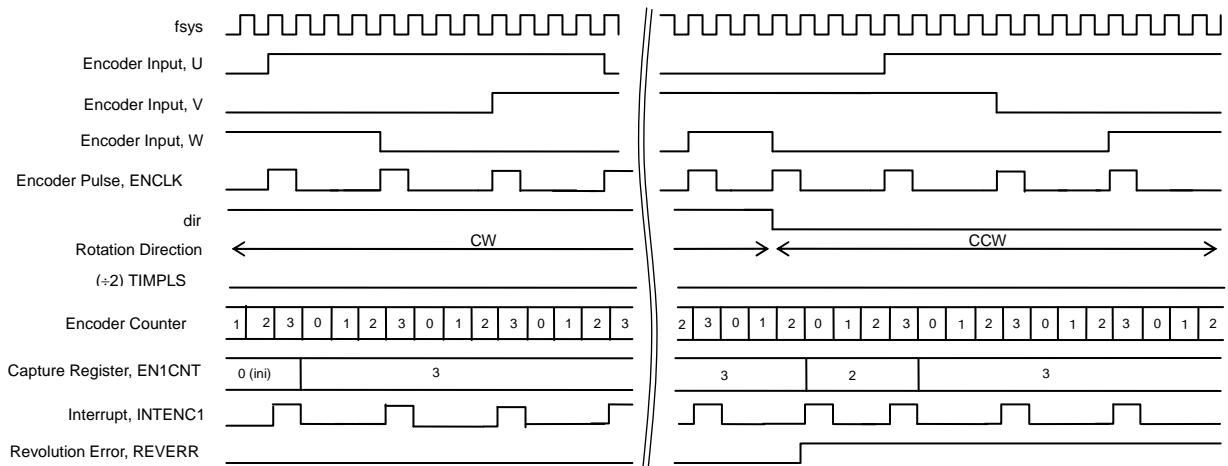
(2) When <P3EN> = 0 (<EN1INT> = 0x0002)
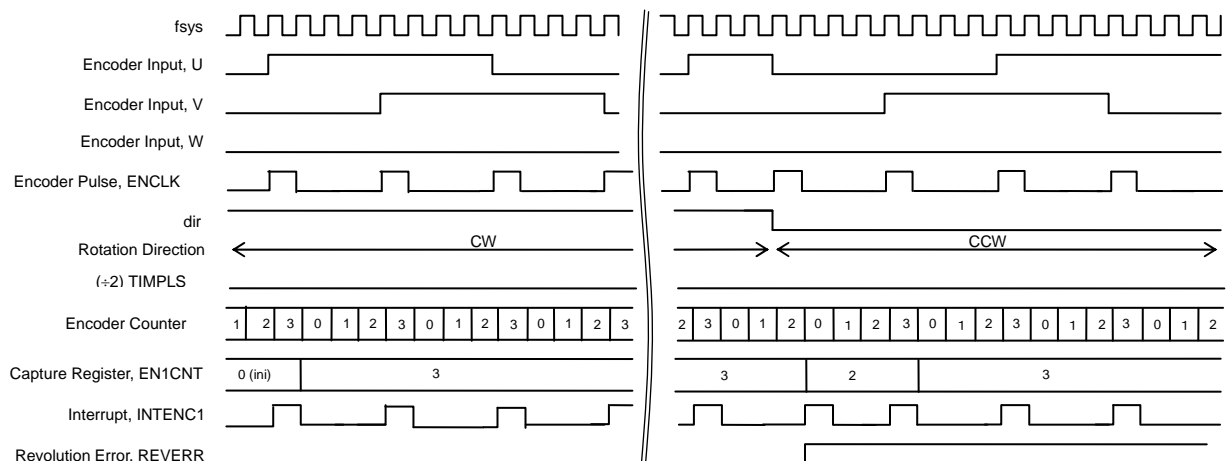
- In Sensor Timer Count mode, the Hall sensor inputs of the TMPM372/3/4 should be connected to the U, V and W channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied_by_4 clock (when <P3EN> = 0) derived from the decoded U and V signals or multiplied_by_6 clock (when <P3EN> = 1) derived from the decoded U, V and W signals.

- The encoder counter always counts up; it is cleared to 0 on ENCLK. When the encoder counter has reached 0xFFFFFF, it wraps around to 0.

- When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.

- ENCLK causes the value of the encoder counter to be captured into the EN1CNT register. The captured counter value can be read out of EN1CNT.

- Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.

- Clearing <ENRUN> to 0 clears <UD> to 0.

- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <EN1INT>.

- <UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.

- <REVERR> is set to 1 when the rotation direction has changed. This bit is cleared to 0 on a read.

- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

19.3.1.4 Timer Mode

(1) When <ZEN> = 1 (<EN1INT> = 0x0006)

(2) When <ZEN> = 0 (<EN1INT> = 0x0006)
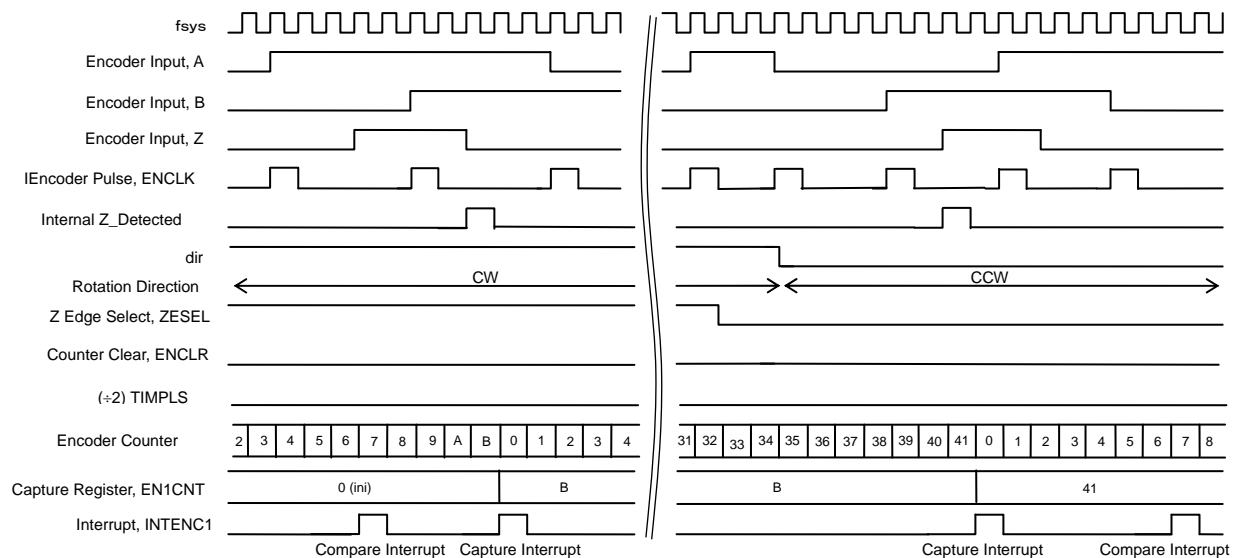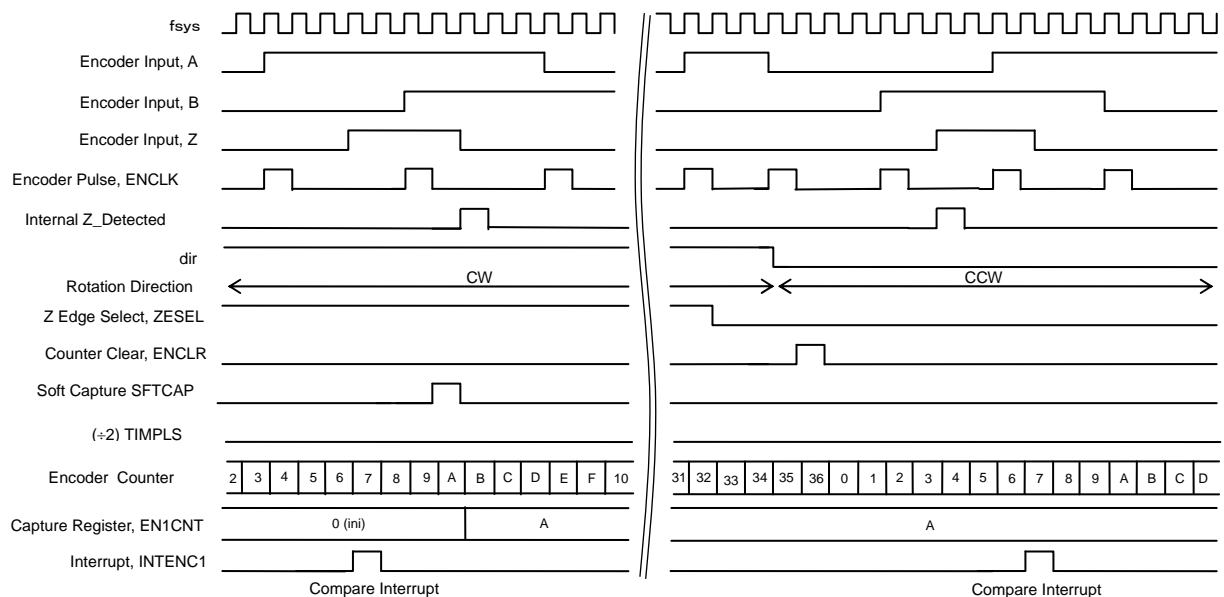
· When <ZEN> = 1, the Z input pin is used as an external trigger. When <ZEN> = 0, no external input is used to trigger the timer.

· The encoder counter always counts up. If <ZEN> = 1, the counter is cleared to 0 on the selected edge of Z (at the internal Z_Detected timing): a rising edge when <ZESEL> = 0 and a falling edge when <ZESEL> = 1. When the encoder counter has reached 0xFFFFFF, it wraps around to 0.

· When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.

· Z_Detected causes the value of the encoder counter to be captured into the ENCNT register. The captured counter value can be read out of ENCNT.

· Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be

captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.

・ <UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.

・ If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <EN1INT>.

・ Clearing <ENRUN> to 0 clears <UD> to 0.

・ The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

19.3.2 Counter Operation and Interrupt Generation When <CMPEN> =1

19.3.2.1 Encoder Mode
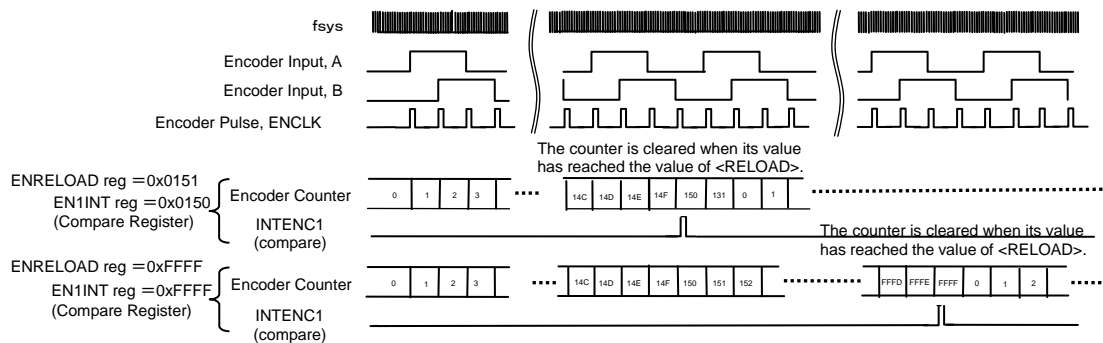
19.3.2.2 Sensor Event Count Mode
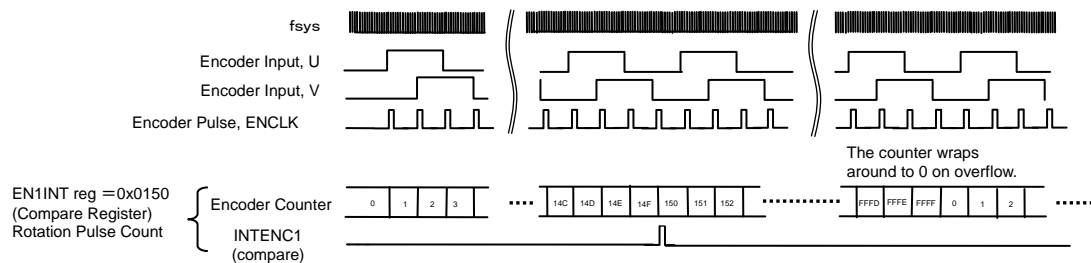
19.3.2.3 Sensor Timer Count Mode
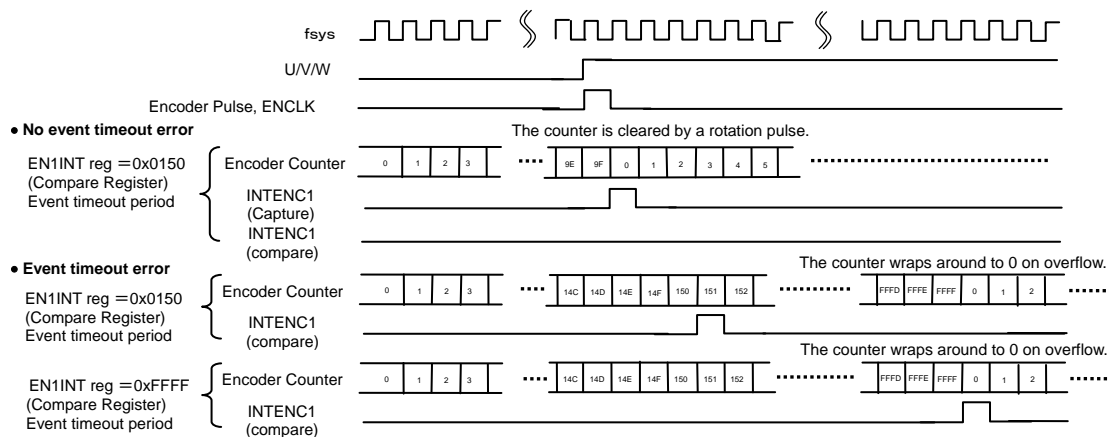
19.3.2.4 Timer Mode
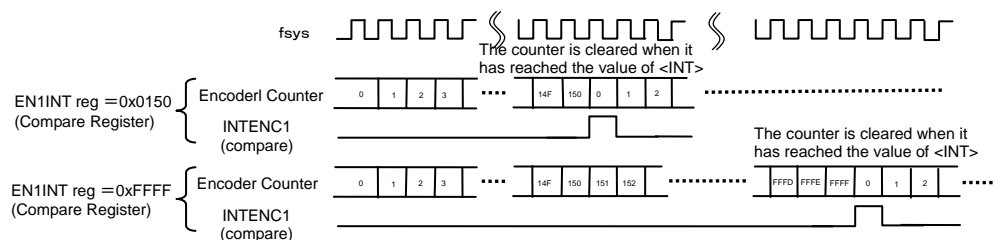
19.3.3 Counter Operation and Interrupt Generation When <CMPEN> = 0

19.3.3.1 Encoder Mode
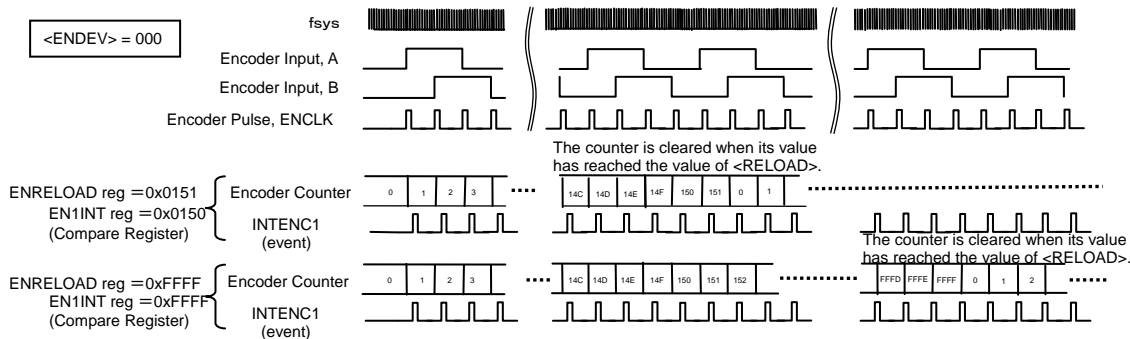
19.3.3.2 Sensor Event Count Mode
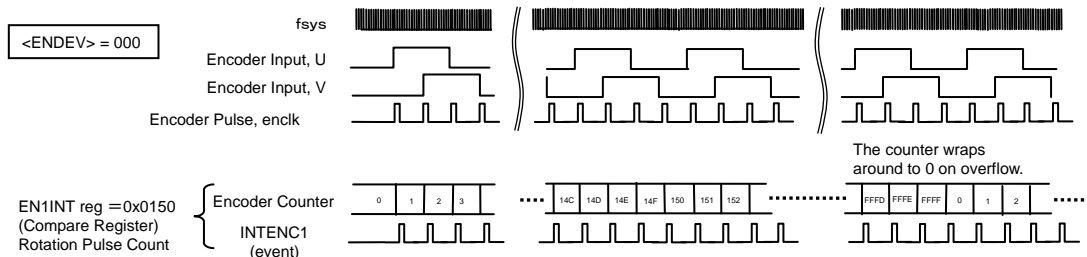
19.3.3.3 Sensor Timer Count Mode
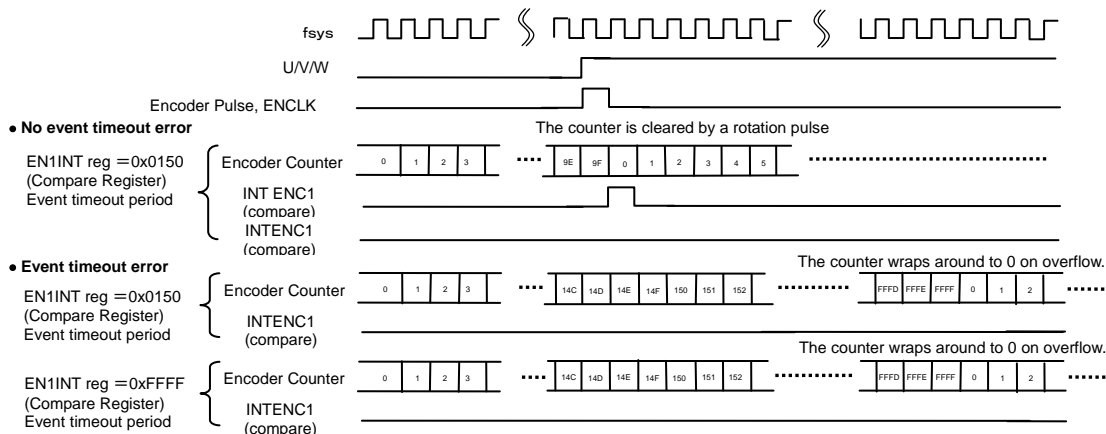
19.3.3.4 Timer Mode
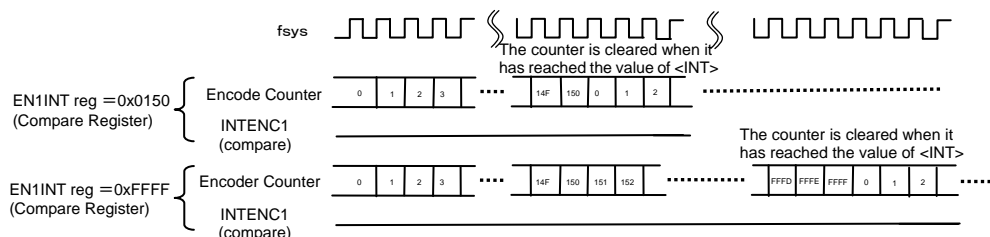
19.3.4   Encoder Rotation Direction

The following diagrams illustrate the phase shifting of the A, B and Z channels.

The ENC can interface with both two-phase (A/B) and three-phase (A/B/Z) encoder inputs. For three-phase encoder inputs, <P3EN> should be set to 1.

(1)   Here are possible combinations of values of the A (U), B (V) and Z (W) signals during CW rotation.

・For two-phase inputs
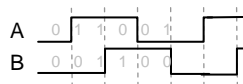
・For three-phase inputs
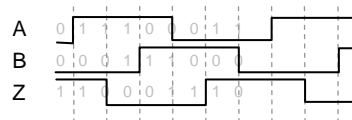
(2)   Here are possible combinations of values of the A (U), B (V) and Z (W) signals during CCW rotation.

・For two-phase inputs
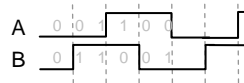
・For three-phase inputs
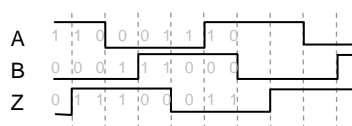
19.3.5   Counter Block

The counter block consists of a 24-bit up/down counter and its control logic.

19.3.5.1 Overview

The counter is configured as an up-counter or a down-counter, cleared and reloaded with a programmed value, according to the selected operating mode.

Table 19-4 summarizes how the counter is controlled.

Table 19-4 Counter Control

Operating Mode <MODE1:0>	<ZEN>	<P3EN>	Input Pins	Count	Up/Down	Counter Clear Conditions	Counter Reload Conditions	Counter Range (Reload Value)
Encoder Mode 00	0	0	A, B	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=<RELOAD>	-	0x0000 thru. <RELOAD>
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
	1		A, B, Z	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=<RELOAD> [3] Z trigger	-	
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
Sensor Event Count Mode 01	0	0	U, V	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=16'hFFFF	-	0x0000 thru. 0xFFFF
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
		1	U, V, W	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFF	-	
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
Sensor Timer Count Mode 10	0	0	U, V	fsys	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFFFF [3] Encoder pulse (ENCLK)		0x000000 thru. 0xFFFFFF
		1	U, V, W	fsys	Up			
Timer Mode 11	0	X	-	fsys	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFFFF [3] counter=<EN1NT>		0x000000 thru. 0xFFFFFF
	1		Z	fsys	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFFFF [3] counter=<EN1INT> [4] Z trigger		

Note: Clearing <ENRUN> to 0 does not clear the counter.
Setting <ENRUN> to 1 again causes the counter to restart from the current count.
The counter should be cleared by software setting <ENCLR> to 1.

19.3.6   Interrupts

The ENC has these interrupts: event (divided-clock/capture) interrupt, event timeout interrupt, timer compare interrupt and capture interrupt.

19.3.6.1 Overview

When <INTEN> = 1, the ENC generates interrupt requests, based on the counter value and the detection of a encoder pulse.

There are six interrupt sources, depending on the operating mode, and the settings of <CMPEN> and <ZEN>, as shown in Table 19-5.

Table 19-5 Interrupt Sources

	Interrupt Source	Description	Operating Mode	Interrupt Generation	Status Flag
1	Event counter interrupt	When <CMPEN> = 1, the encoderl counter counts events (encoder pulses). When it has reached the value programmed in <EN1INT>, an interrupt occurs.	Encoder mode and Sensor Event Count modes	When <INTEN> = 1 *and* <CMPEN> = 1	<CMP>
2	Event interrupt (Divided clock pulse)	An interrupt occurs on each divided clock pulse, which is derived by dividing the encoder pulse by a factor programmed in <ENDEV>.		When <INTEN> = 1	None
3	Event interrupt (Capture interrupt)	An interrupt occurs to indicate that an event (encoder pulse) has occurred, causing the counter value to be captured.	Sensor Timer Count mode	When <INTEN> = 1	None
4	Event timeout 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 <EN1INT>, an interrupt occurs.		When <INTEN> = 1 *and* <CMPEN> = 1	<CMP>
5	Timer compare interrupt	When <CMPEN> = 1, an interrupt occurs when the timer has reached the value programmed in <EN1INT>.	Timer mode	When <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).		When <INTEN> = 1	None

In Sensor Timer Count mode and Timer mode, the value of the encoder counter can be captured into the ENCNT register.

The captured counter value can be read out of the ENCNT register.

In Sensor Timer Count mode, the value of the encoder counter is captured into the ENCNT register upon occurrence of an event (encoder pulse). The counter value can also be captured by writing a 1 to <SFTCAP>.

In Timer mode, the counter value can be captured by writing a 1 to <SFTCAP>. If <ZEN> is set to 1, the counter value can also be captured by an edge of the Z signal input selected via <ZESEL>.

20.   Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

20.1 Flash Memory

20.1.1 Features

1) Memory capacity

The TMPM372/3/4FW 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. The TMPM372/3/4FW contains 64 words in a page.
Page writing requires 1.25ms (typical) regardless of number of words.
A block erase requires 0.1 sec. (typical).
The following table shows write and erase time per chip.

Product Name	Memory Size	Block Configuration				# of Words	Write Time	Erase Time
		128KB	64KB	32KB	16KB			
TMPM372/3/4FW	128KB	0	1	1	2	64	0.64sec	0.2sec

**(Note) The above values are theoretical values not including data transfer time.
The write time per chip depends on the write method to be used by the user.**

3) Programming method

The onboard programming mode is available for the user to program (rewrite) the device while it is mounted on the user's board.

・The onboard programming mode

   3-1) User boot mode
    The user's original rewriting method can be supported.

   3-2) Single boot mode
    The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

4) Rewriting method

The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if the user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

5) Protect/Security function

This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See chapter 20 for details of ROM protection and security function.

20.1.2 Block Diagram of the Flash Memory Section
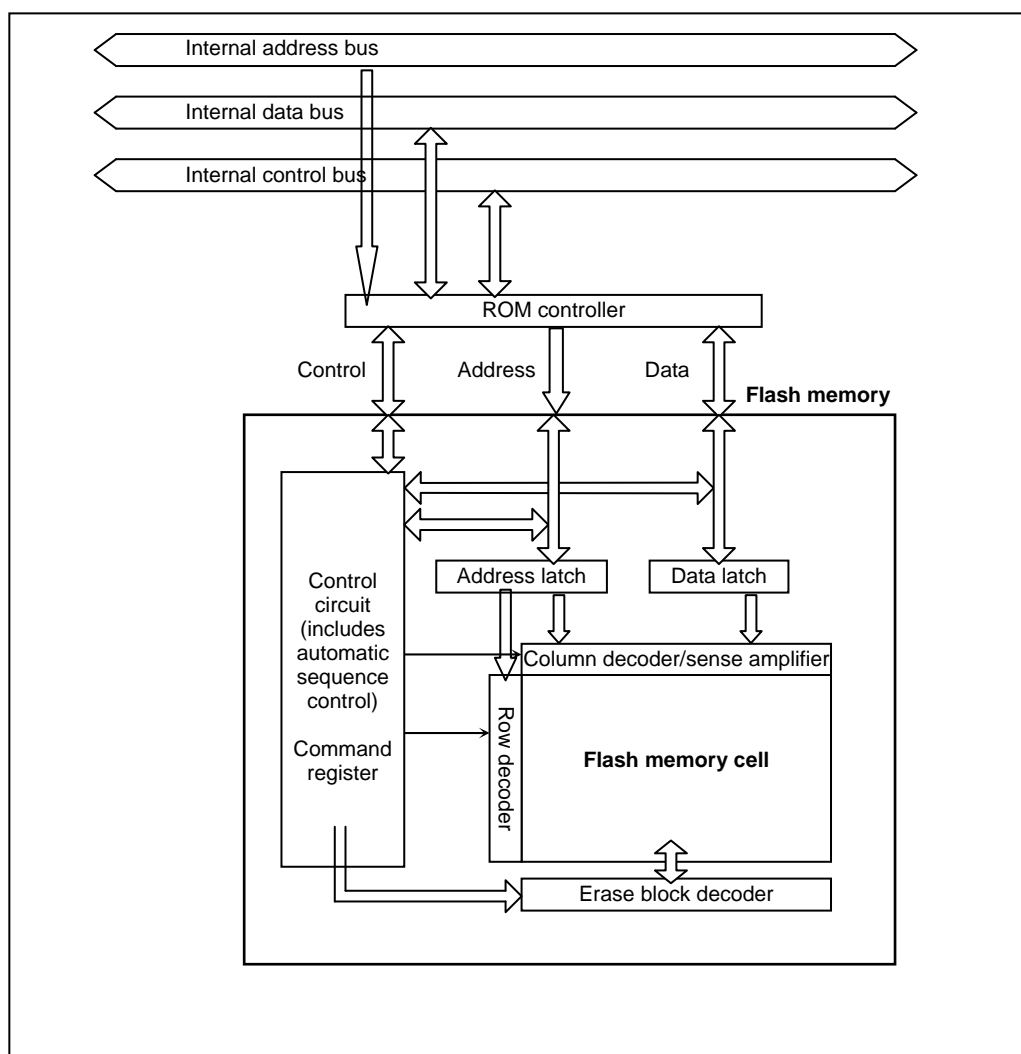
Fig 20-1 Block Diagram of the Flash Memory Section

20.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 20-1 Operation Modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode.
	User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "PA4" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ (PF0) pin while the device is in reset status.

Table 20-2 Operation Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PF0)
Single chip mode	$0 \rightarrow 1$	1
Single boot mode	$0 \rightarrow 1$	0
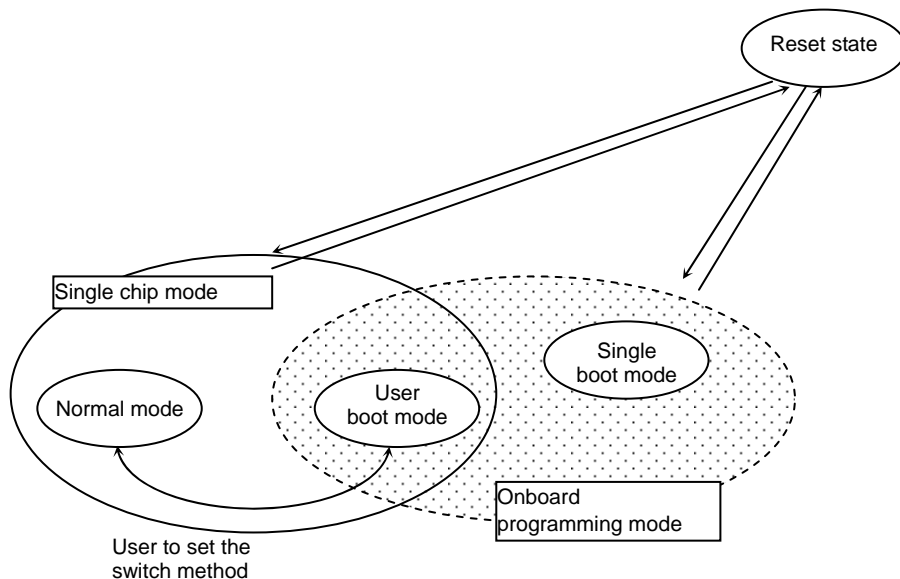
Fig 20-2　Mode Transition Diagram

20.2.1　Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the $\overline{\text{RESET}}$ input is held at "0" for a minimum duration of 12 system clocks (0.15 µs with 80MHz operation; the "1/1" clock gear mode is applied after reset).

(Note 1) Regarding power-on reset of devices with internal flash memory; for devices with internal flash memory, it is necessary to apply "0" to the $\overline{\text{RESET}}$ inputs upon power on for a minimum duration of 3.2 milliseconds regardless of the operating frequency.

(Note 2) While flash auto programming or deletion is in progress, at least 0.5 microseconds of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

20.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of TMPM372/3/4 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. All the interruption including a non-maskable are inhibited at 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 20.3 On-board Programming of Flash Memory (Rewrite/Erase).

User Boot Mode

   (1-A)   Method 1: Storing a Programming Routine in the Flash Memory

(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 TMPM372/3/4 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 TMPM372/3/4 flash memory to either the TMPM372/3/4 on-chip RAM or external memory device.
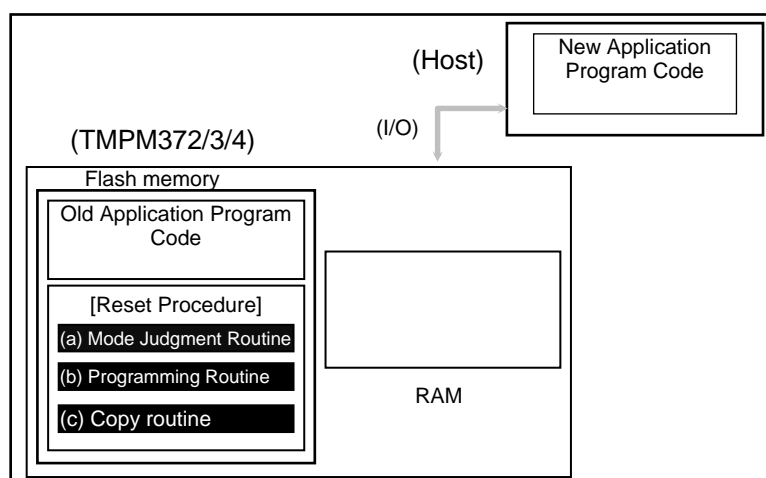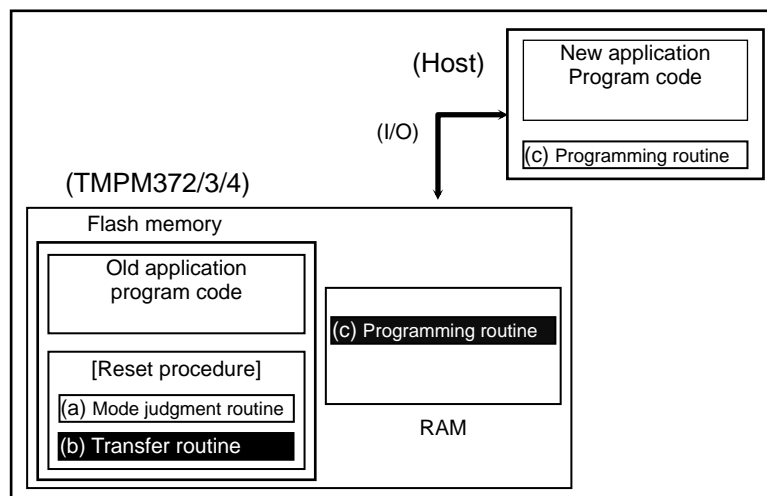

(Step-2)

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM372/3/4 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode.)
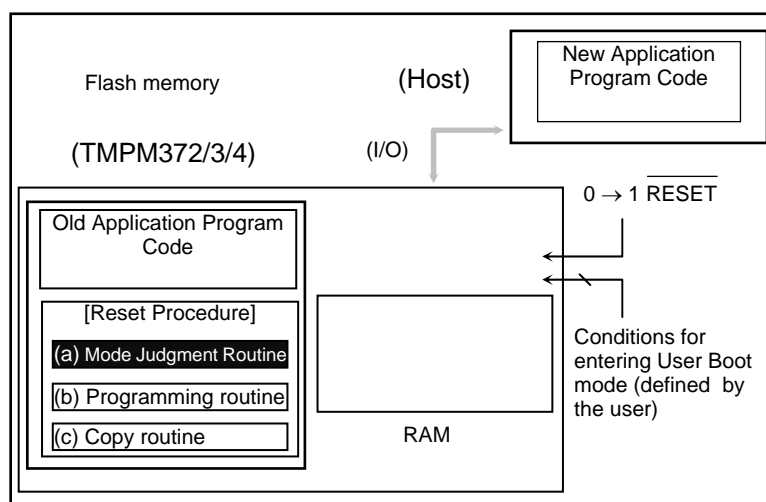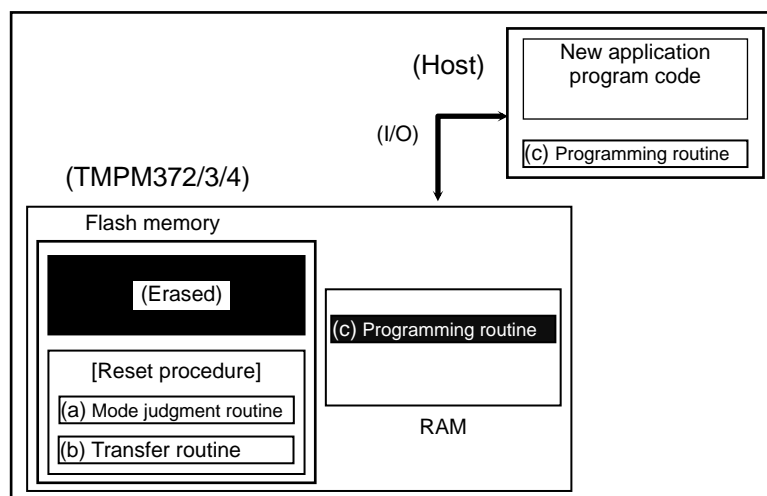
(Step-3)

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM372/3/4 on-chip RAM.
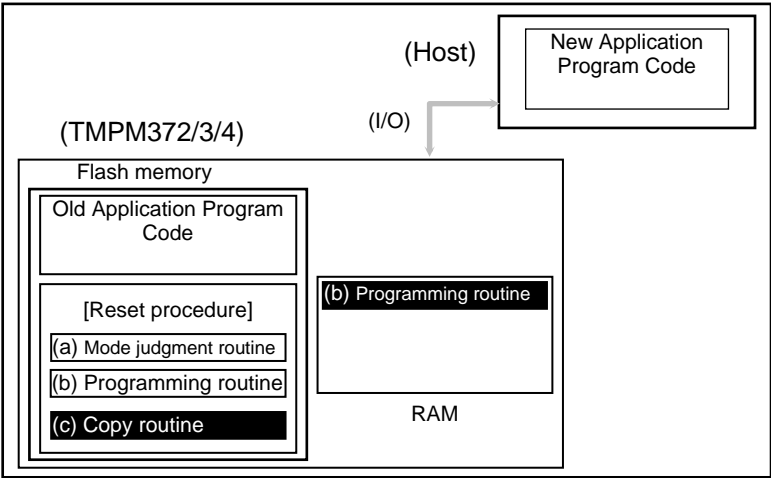

(Step-4)

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.
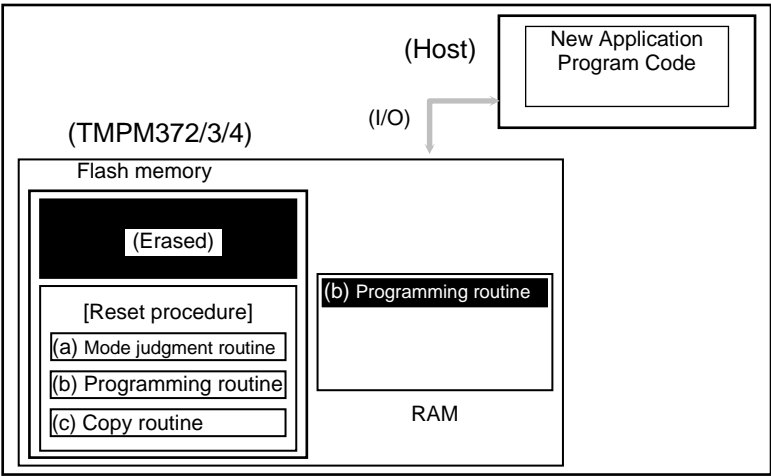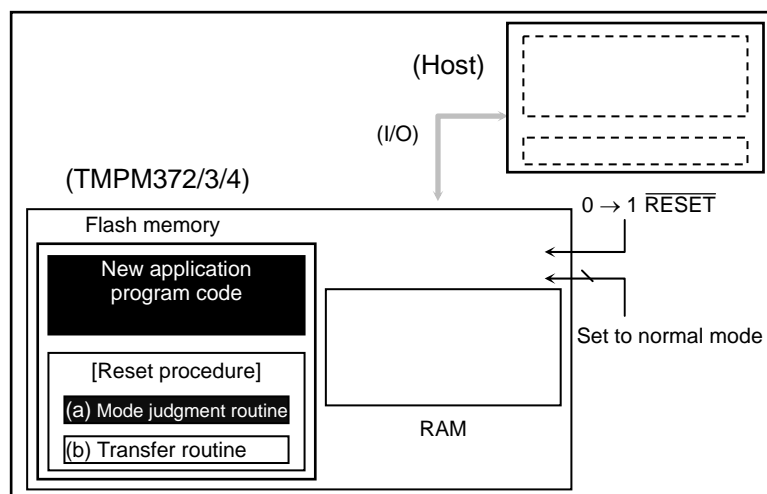
(Step-5)

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.
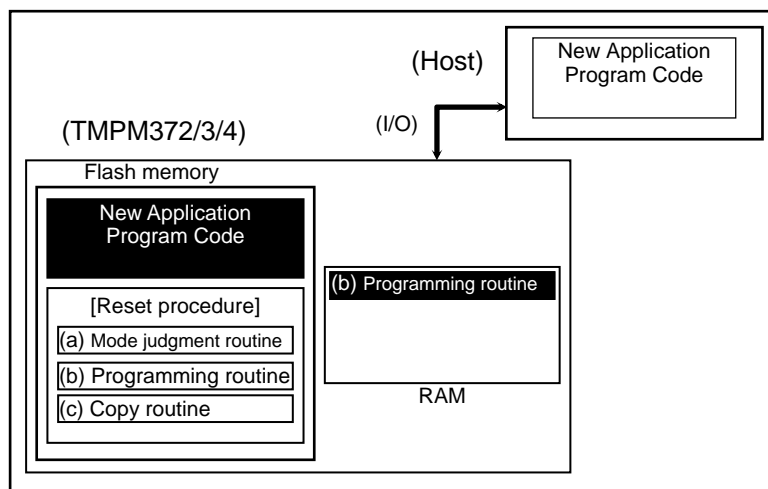
(Step-6)

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM372/3/4. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.
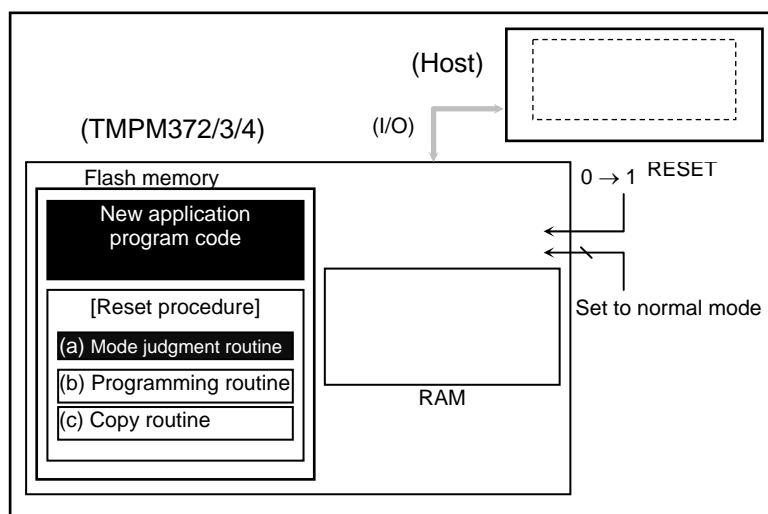
(1-B) Method 2: Transferring a Programming Routine from an External Host

(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 TMPM372/3/4 on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode
(b) Transfer routine: Code to download new program code from a host controller

Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory
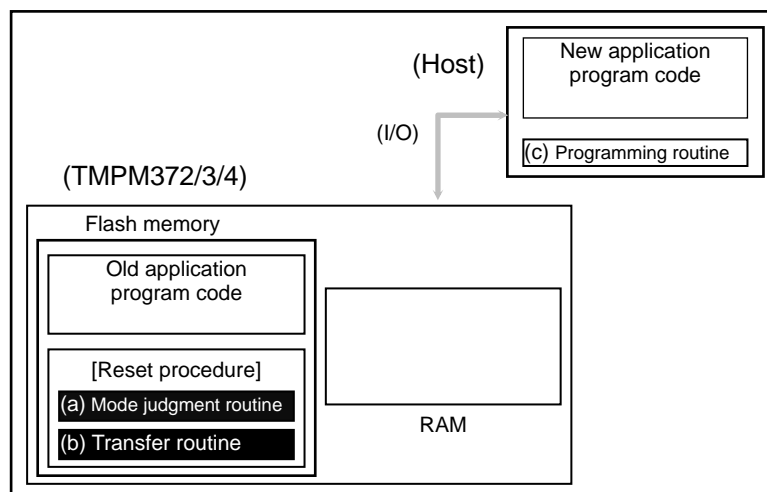
(Step-2)

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM372/3/4 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode).
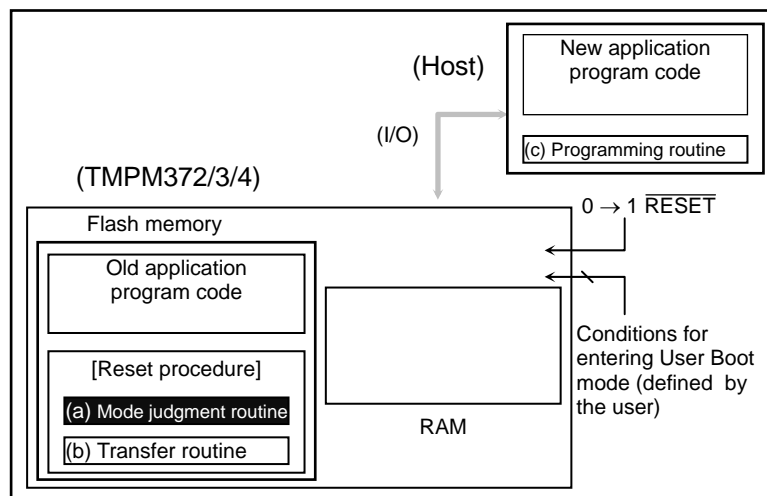
(Step-3)

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM372/3/4 on-chip RAM.

```
                                                 (Host)    ┌──────────────────────┐
                                                           │  New application     │
                                                           │  Program code        │
                                                           │ ┌──────────────────┐ │
                                                (I/O)       │ │(c) Programming   │ │
                                                     ──────►│ │    routine       │ │
                                                           └──────────────────────┘
        (TMPM372/3/4)
        ┌──────────────────────────────────────────────────────────────┐
        │  Flash memory                                                  │
        │  ┌──────────────────┐        ┌──────────────────────────┐      │
        │  │ Old application  │        │ (c) Programming routine  │      │
        │  │ program code     │        │                          │      │
        │  └──────────────────┘        │                          │      │
        │  ┌──────────────────┐        │                          │      │
        │  │[Reset procedure] │        └──────────────────────────┘      │
        │  │(a) Mode judgment │                RAM                       │
        │  │    routine       │                                          │
        │  │(b) Transfer routine                                         │
        │  └──────────────────┘                                          │
        └──────────────────────────────────────────────────────────────┘
```

(Step-4)

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.

```
                                                 (Host)    ┌──────────────────────┐
                                                           │  New application     │
                                                           │  program code        │
                                                           │ ┌──────────────────┐ │
                                                (I/O)       │ │(c) Programming   │ │
                                                     ──────►│ │    routine       │ │
                                                           └──────────────────────┘
        (TMPM372/3/4)
        ┌──────────────────────────────────────────────────────────────┐
        │  Flash memory                                                  │
        │  ┌──────────────────┐        ┌──────────────────────────┐      │
        │  │    (Erased)      │        │ (c) Programming routine  │      │
        │  └──────────────────┘        │                          │      │
        │  ┌──────────────────┐        │                          │      │
        │  │[Reset procedure] │        └──────────────────────────┘      │
        │  │(a) Mode judgment │                RAM                       │
        │  │    routine       │                                          │
        │  │(b) Transfer routine                                         │
        │  └──────────────────┘                                          │
        └──────────────────────────────────────────────────────────────┘
```

(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. Once programming is complete, turn on the protection of that flash block.
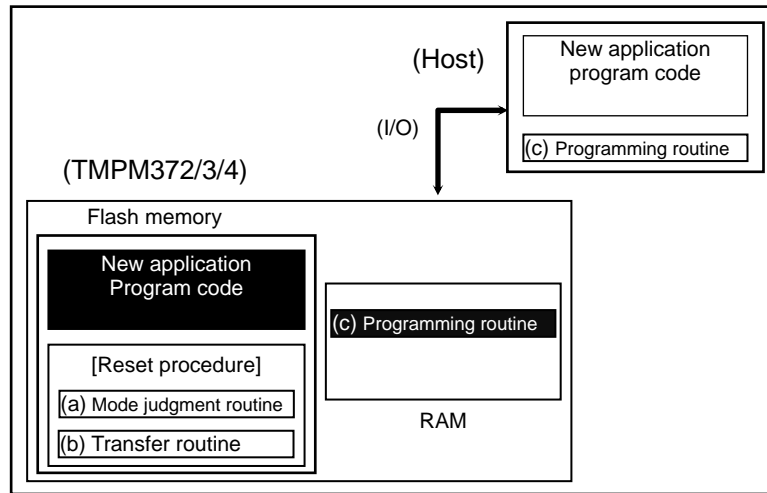
(Step-6)

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM372/3/4. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.

20.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM372/3/4 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 TMPM372/3/4 is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM372/3/4 on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory.

Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is checked before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted.

As in the case of User Boot mode, all interrupts including the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is being erased or programmed. In SingleBoot mode, the boot-ROM programs are executed in Normal mode.

Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

Single Boot Mode

(2-A) Using the Program in the On-Chip Boot ROM

(Step-1)

The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.
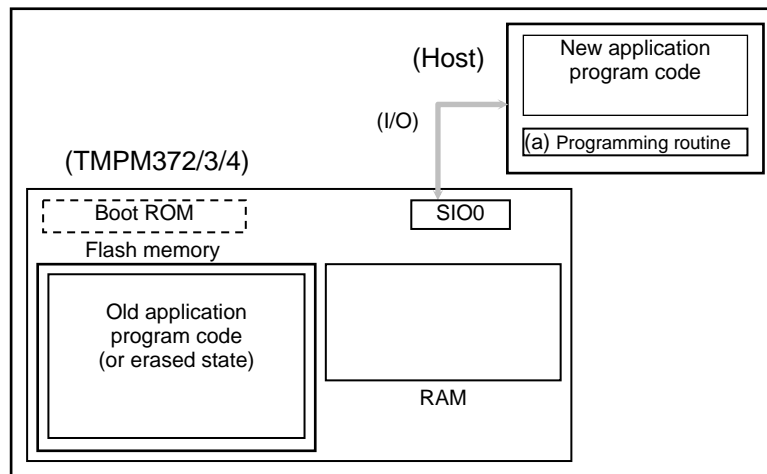
(Step-2)

Cancel the reset of the TMPM372/3/4 by setting the Single Boot mode pin to "0", so that the CPU re-boots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is first compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFFFF).
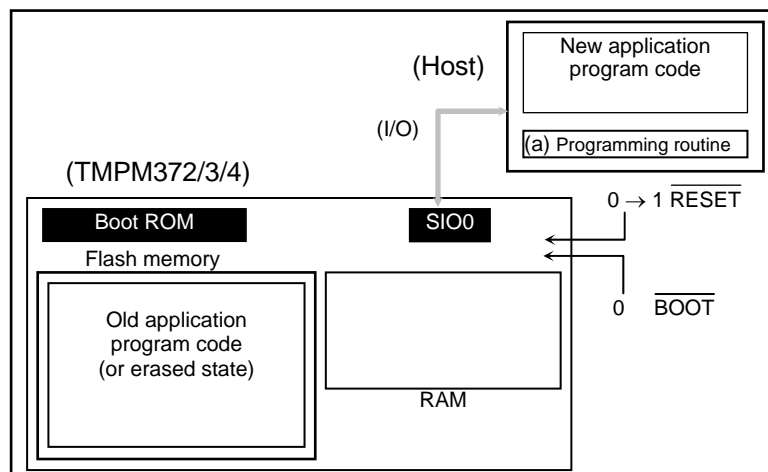
(Step-3)

If the password was correct, the boot program downloads, via the SIO0, the programming routine (a) from the host controller into the on-chip RAM of the TMPM372/3/4. The programming routine must be stored in the address range 0x2000_0400 to the end address of RAM.
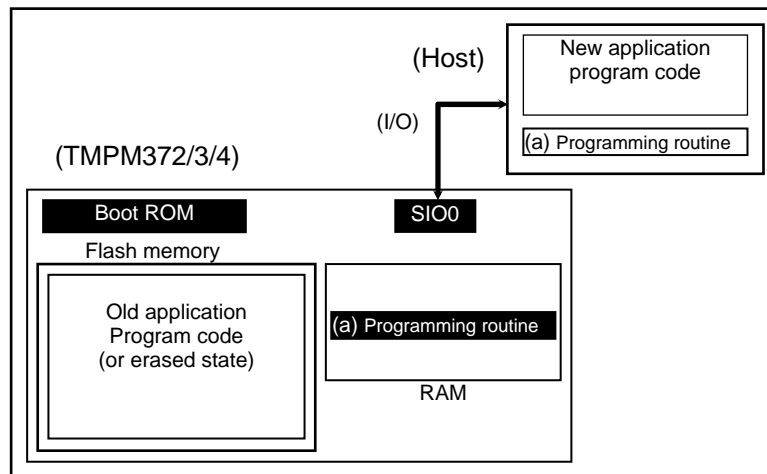
(Step-4)

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.
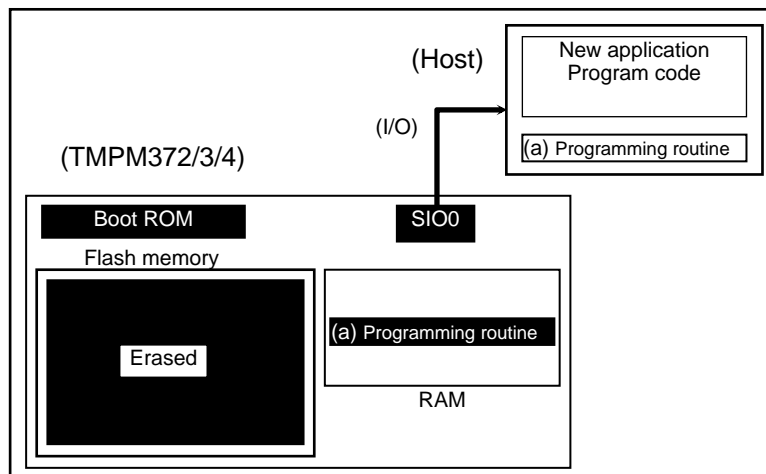
(Step-5)

Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. Once programming is complete, protection of that flash block is turned on. It is not allowed to move program control from the programming routine (a) back to the boot ROM.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.
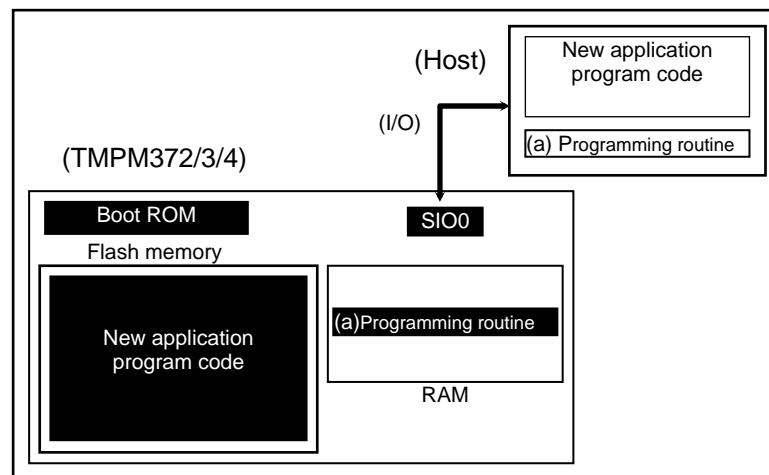
(Step-6)

When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMPM372/3/4 re-boots in Single-Chip (Normal) mode to execute the new program.
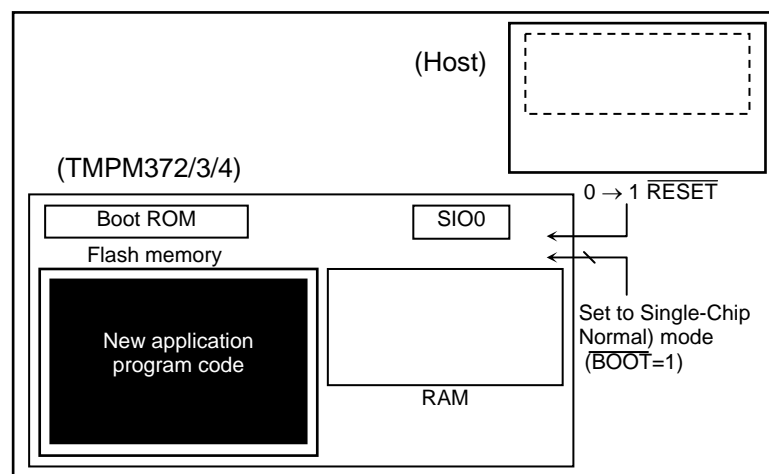
(1) Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM372/3/4 with Single Boot mode following the configuration shown below.

$$\overline{BOOT}\ (PF0) \qquad = 0$$

$$\overline{RESET} \qquad\qquad = 0 \rightarrow 1$$

Set the $\overline{RESET}$ input to 0, and set the each $\overline{BOOT}$ (PF0) pins to values shown above, and then release $\overline{RESET}$ (high).

(2) Memory Map

Fig 20-3 shows a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the internal flash memory is mapped to 0x3F80_0000 through 0x3F81_FFFF, and the Internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_0FFF.

Product Name	Flash Size	RAM Size	Flash Address (Single Chip/ Single Boot Mode)	RAM Address
TMPM372/3/4FW	128KB	6KB	0x0000_0000 - 0x0001_FFFF 0x3F80_0000 - 0x3F81_FFFF	0x2000_0000 - 0x2000_17FF
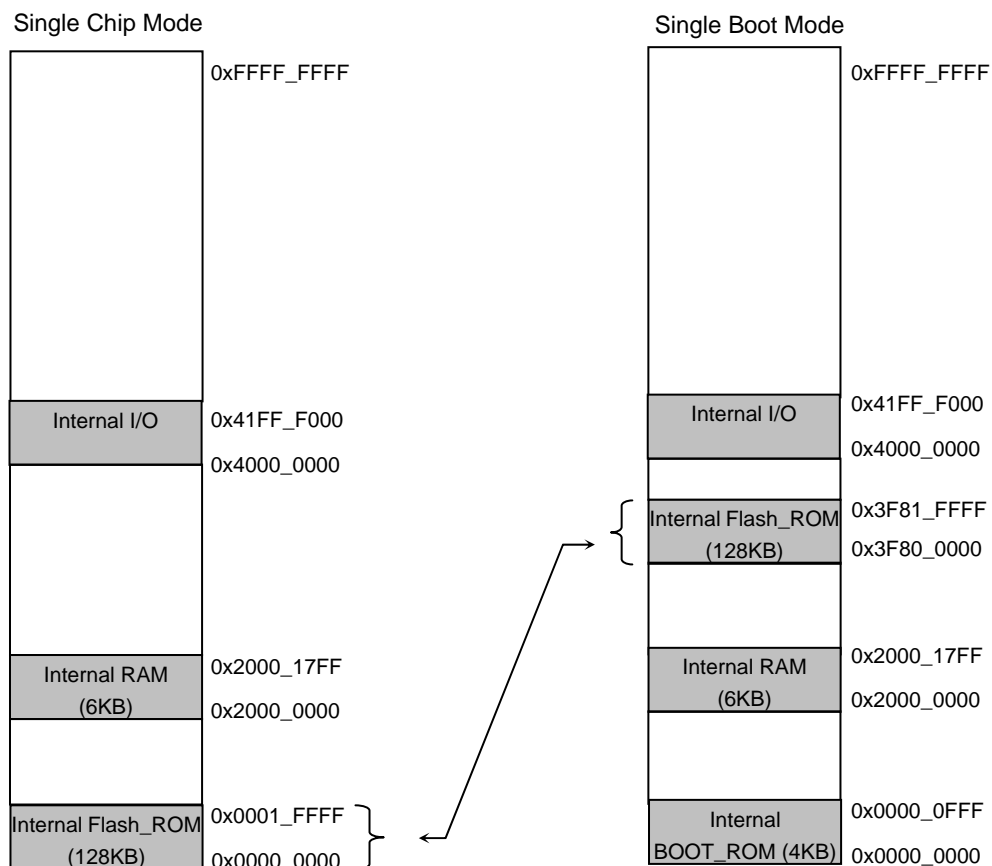
Fig 20-3 Memory Maps for TMPM372/3/4FW

(3) 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 fast
  Data length   : 8 bit
  Parity bits   : None
  STOP bits   : 1 bit
  Baud rate   : Arbitrary baud rate

- I/O interface mode
  Communication channel   : SIO channel 0
  Serial transfer mode   : I/O interface mode, full -duplex, LSB fast
  Synchronization clock (SCLK0)   : Input mode
  Handshaking signal   : PE4 configured as an output mode
  Baud rate   : Arbitrary baud rate

-

Table 20-3 Required Pin Connections

Pins		Interface	
		UART	I/O Interface Mode
Power supply pins	RVDD5	○	○
	AVDD5B	○	○
	DVDD5	○	○
	VOUT15	○	○
	VOUT3	○	○
	AVSS B	○	○
	DVSS	○	○
Mode-setting pin	$\overline{\text{BOOT}}$ (PF0)	○	○
Reset pin	$\overline{\text{RESET}}$	○	○
Communication pins	TXD0(PE0)	○	○
	RXD0(PE1)	○	○
	SCLK0(PE2)	x	○ (Input mode)
	PE4	x	○ (Output mode)

(4) Data Transfer Format

Table 20-4, Table 20-6 and Table 20-7 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to (6) Operation of Boot Program.

Table 20-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x40	Chip and protection bit erase

(5) Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 20-5.

Table 20-5 **Restrictions in Single Boot Mode**

Memory	Details
Internal RAM	BOOT ROM is mapped from 0x2000_0000 to 0x2000_03FF. 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. TMPM372/3/4FW : 0x3F81_FFF0 to 0x3F81_FFFF

Table 20-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM372/3/4FW	Baud rate	Data Transferred from the TMPM372/3/4FW 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 0xN1 -Communication error 0xN8
	5 byte - 16 byte	Password sequence (12 bytes) 0x3F81_FFF4 to 0x3F81_FFFF		—
	17 byte	Check SUM value for bytes 5 - 16		-
	18 byte	-		ACK for the checksum byte (Note 2) -Normal acknowledge 0x10 -Negative acknowledge 0xN1 -Communication error 0xN8
	19 byte	RAM storage start address 31 - 24		-
	20 byte	RAM storage start address 23 - 16		-
	21 byte	RAM storage start address 15 - 8		-
	22 byte	RAM storage start address 7 - 0		-
	23 byte	RAM storage byte count 15 - 8		-
	24 byte	RAM storage byte count 7 - 0		-
	25 byte	Check SUM value for bytes 19 - 24		-
	26 byte	-		ACK for the checksum byte (Note 2) -Normal acknowledge 0x10 -Negative acknowledge 0xN1 -Communication error 0xN8
	27 byte ~ m byte	RAM storage data		-
	m + byte	Checksum value for bytes 27 - m		-
	m + byte	-		ACK for the checksum byte (Note 2) -Normal acknowledge 0x10 -Negative acknowledge 0xN1 -Communication error 0xN8
RAM	m + 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.

Table 20-7 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM372/3/4FW	Baud rate	Data Transferred from the TMPM372/3/4FW to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode 0x86 For I/O Interface mode 0x30	Desired baud rate (Note 1)	—
	2 byte	—		ACK for the serial operation mode byte For UART mode -Normal acknowledge 0x86 For I/O Interface mode -Normal acknowledge 0x30 (The boot program aborts if the baud rate can not be set correctly.)
	3 byte	Command code (0x40)		—
	4 byte	—		ACK for the command code byte (Note 2) -Normal acknowledge 0x40 -Negative acknowledge 0xN1 -Communication error 0xN8
	5 byte	Chip erase command code (0x54)		—
	6 byte	—		ACK for the command code byte (Note 2) -Normal acknowledge 0x54 -Negative acknowledge 0xN1 -Communication error 0xN8
	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 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.

(6) Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these two commands, of which the details are provided on the following subsections. The addresses described in this section are the virtual unless otherwise noted.

1. RAM Transfer command

The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the range from 0x2000_0400 to the end address of RAM, whereas the boot program area (0x2000_0000～0x2000_03FF) is unavailable. The user program starts at the assigned RAM address.

The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 20.3.

Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

Note: If a password is set to 0xFF (erased data), it is difficult to protect data securely to due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

2. Chip and Protection Bit Erase command

This command erases the entire area of the flash memory automatically without verifying a password. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. When the command is completed, the SECBIT <SECBIT> bit is set to "1".

This command serves to recover boot programming operation when a user forgets the password. Therefore password verification is not executed.

1)   RAM Transfer Command (See Table 20-6)

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 Determination of a Serial Operation Mode described later. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the HSC0MOD register is cleared.

   - To communicate in UART mode
     Send, from the controller to the target board, 86H in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the first byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.

   - To communicate in I/O Interface mode
     Send, from the controller to the target board, 0x30 in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

     In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip's operating frequency is high, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate. When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the third byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (bit 3, 0xN8).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the first byte: 0x86 for UART mode and 0x30 for I/O Interface mode.

   UART mode
   If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the SC0BRCR and sends back 0x86 to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication. Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. The boot program sets the RXE bit in the SC0MOD0 register to enable reception (1) before loading the SIO transmit buffer with 0x86.

   - I/O Interface mode
     The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 0x30 to

the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller should take it as a go-ahead. The controller must then deliver the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 0x30.

3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (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 20-4, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in a later section "Password". If the 3rd byte is not a valid command, the boot program sends back 0xN1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5th byte and the smallest address in the designated area. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Area
TMPM372/3/4FW	0x3F81_FFF4 – 0x3F81_FFFF

    Note: Regardless using or not using Single Boot mode, password must be written in password area.

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 18H (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., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

Finally, the RAM Transfer routine examines the result of the password verification. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31–24 of the address and the 22nd byte corresponds to bits 7–0 of the address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15–8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7–0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 25th bytes must result in 00H (with the carry dropped). If it is not 00H, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- The RAM storage start address must be within the range of 0x2000_0400 to the end address of RAM.

When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMPM372/3/4FW. Storage begins at the address specified by the 19th–22nd bytes and continues for the number of bytes specified by the 23rd–24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes.

First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there was a receive error, the RAM Transfer routine sends back 18H (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., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes.

2) Chip and Protection Bit Erase command (See Table 20-7)

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. 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. 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 x8H (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 20-4, 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 0xN1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The 5th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).

5. 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 0xN8 (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 Show Flash Memory Sum 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 0xN1 (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. 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 9th byte is the next command code.

3) Acknowledge Responses

The boot program represents processing states with specific codes. Table 20-8 to Table 20-11 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always 0. Receive error checking is not done in I/O Interface mode.

Table 20-8 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

(Note) If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.

Table 20-9 ACK Response to the Command Byte

Return Value	Meaning
0xN8 (See Note)	A receive error occurred while getting 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 20-10 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. It is 1 (N＝RAM transfer command data [7:4]) when password error occurs.

Table 20-11 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.

4)   Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 the desired baud rate. Fig 20-4 shows the waveforms for the first byte.
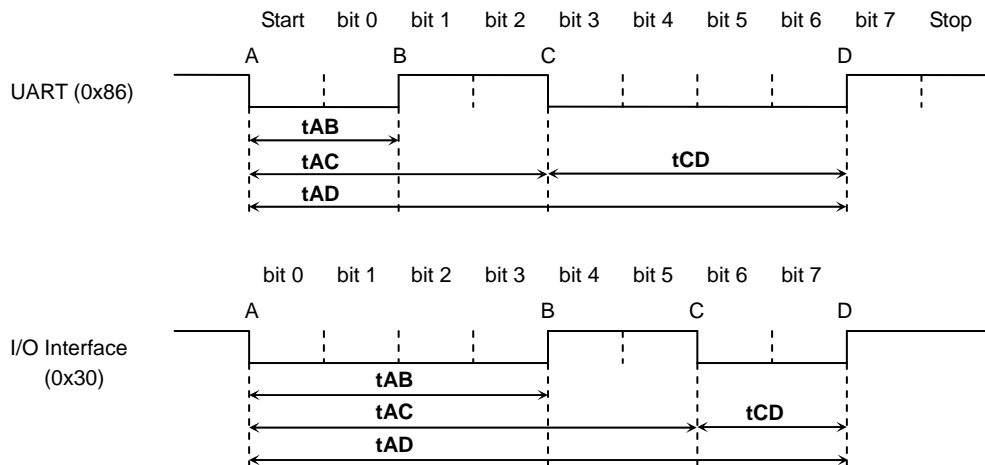

Fig 20-4 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Fig 20-4 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Fig 20-5 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is

greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine
the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater
than tCD and SIO is selected by the resolution of the operation mode determination, the second byte
code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to
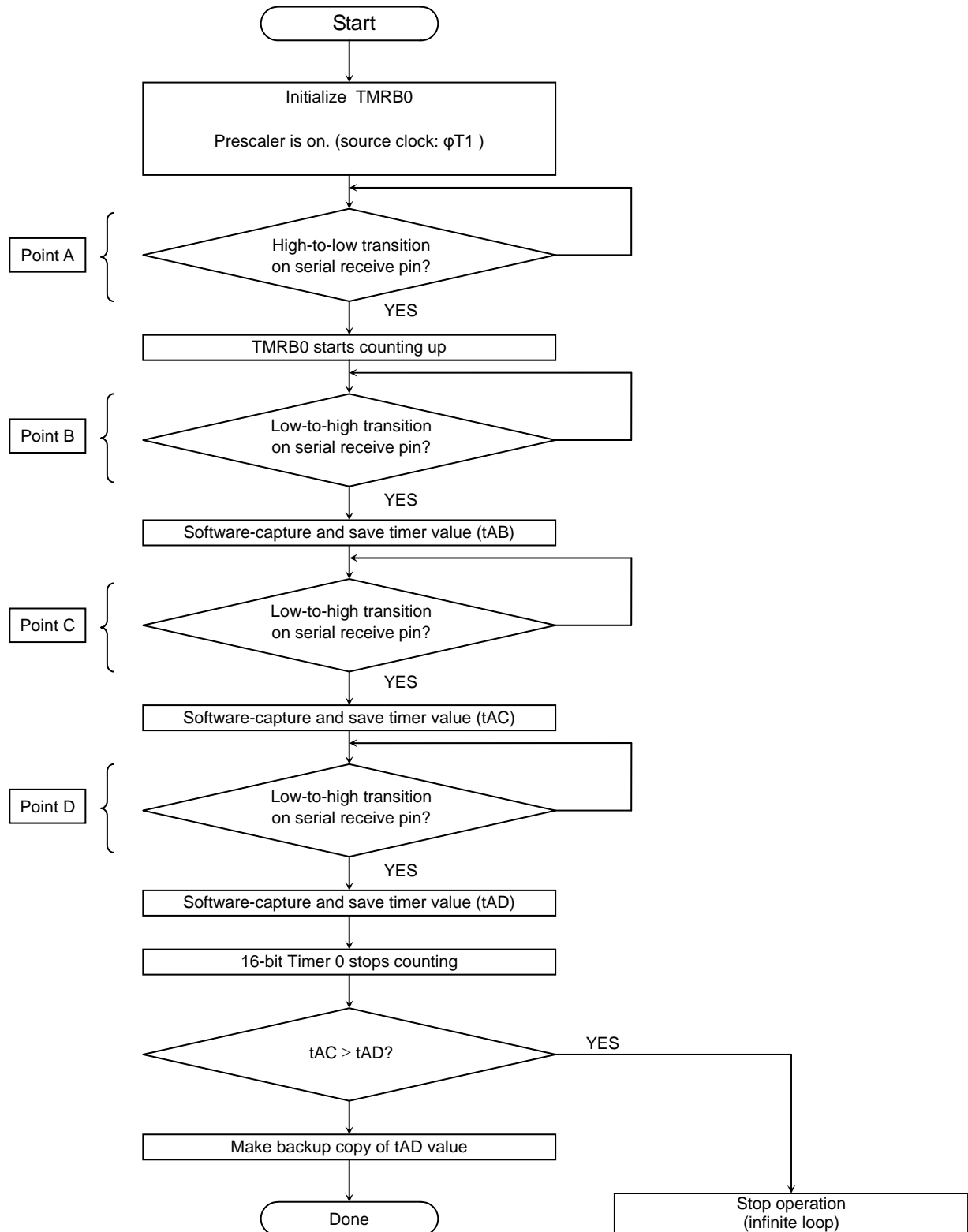determine I/O interface mode is described as 0x30).

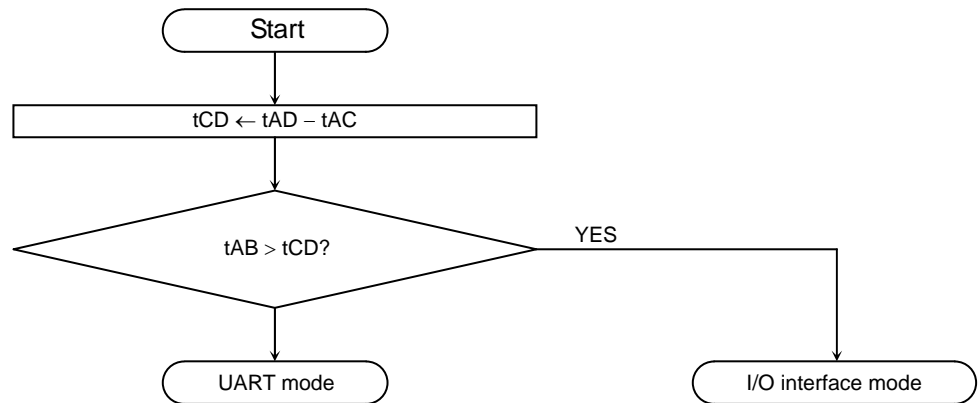Fig 20-5 Serial Operation Mode Byte Reception Flow

Fig 20-6 Serial Operation Mode Determination Flow

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.

Product name	Area
TMPM372/3/4FW	0x3F81_FFF4 - 0x3F81_FFFF

Note: If a password is set to 0xFF (erased data), it is difficult to protect data securely to 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 Fig 20-7.

In this case, the boot program returns an error acknowledge (0x11) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all 0xFFs.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

The password verification is performed even if the security function is enabled.
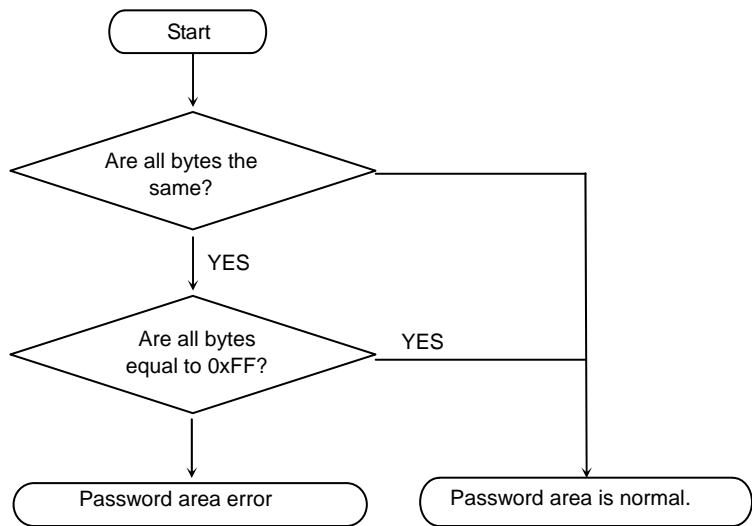
Fig 20-7 Password Area Verification Flow

6) Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together, dropping the carries, and taking the two's complement of the total sum. The Show Flash Memory Sum command and the Show Product Information command perform the checksum calculation. The controller must perform the same checksum operation in transmitting checksum bytes.

Example) Assume the Show Flash Memory Sum command provides the upper and lower bytes of the sum as 0xE5 and 0xF6. To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together
        0xE5 + 0xF6 = 0x1DB

Take the two's complement of the sum, and that is the checksum byte.
        0 − 0xDB = 0x25

(7) General Boot Program Flowchart

Fig 20-8 shows an overall flowchart of the boot program.
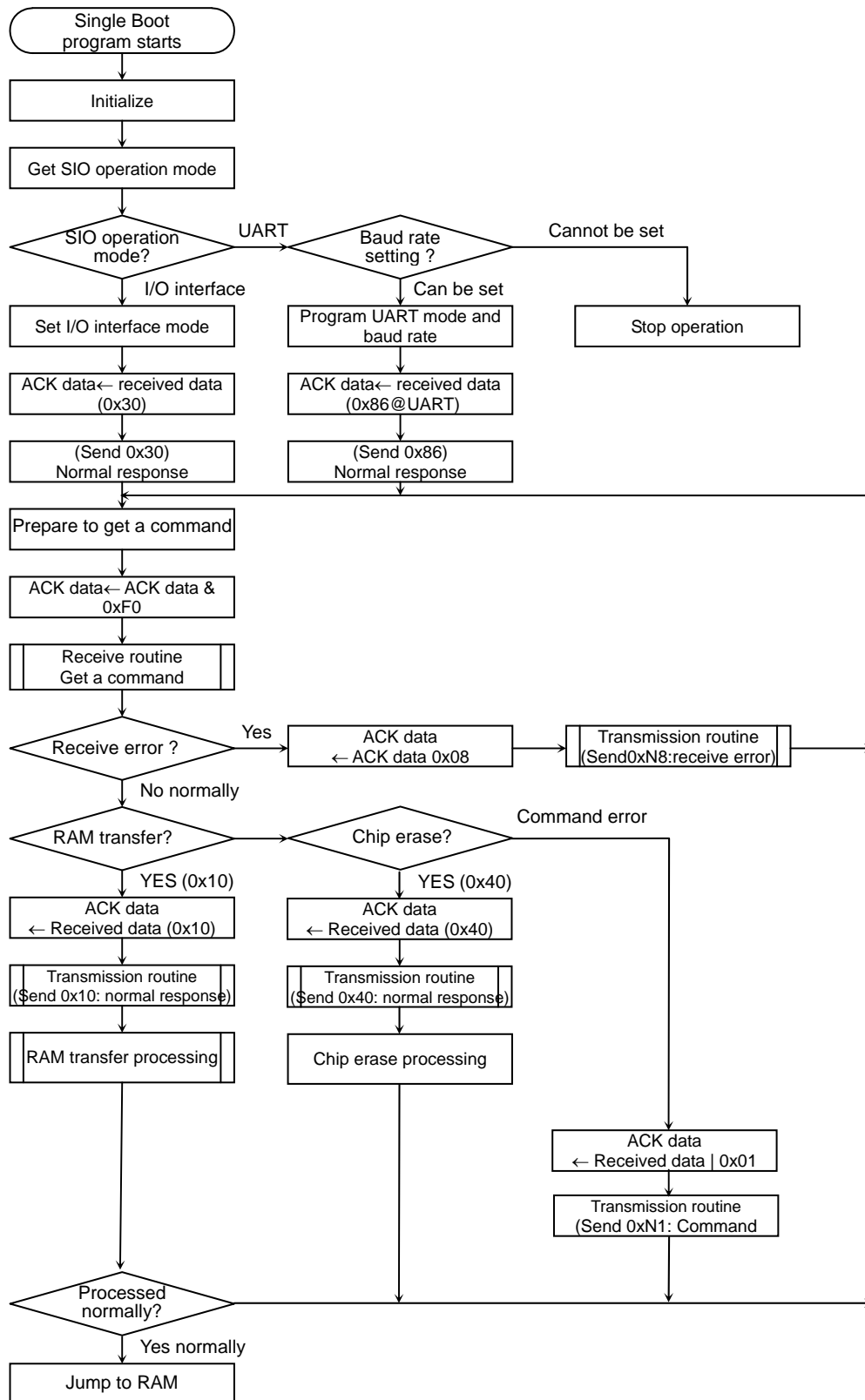

Fig 20-8 Overall Boot Program Flow

20.3 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 or from an external memory device after shifting to the user boot mode.

20.3.1 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 20-12 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erases the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	By writing a 4-bit protection code, the write or erase function can be individually inhibited for each block.

Note that addressing of operation commands is different from the case of standard commands due to the specific interface arrangements with the CPU. Also note that the flash memory is written in 32-bit blocks. So, 32-bit (word) data transfer commands must be used in writing the flash memory.

(1) Block configuration

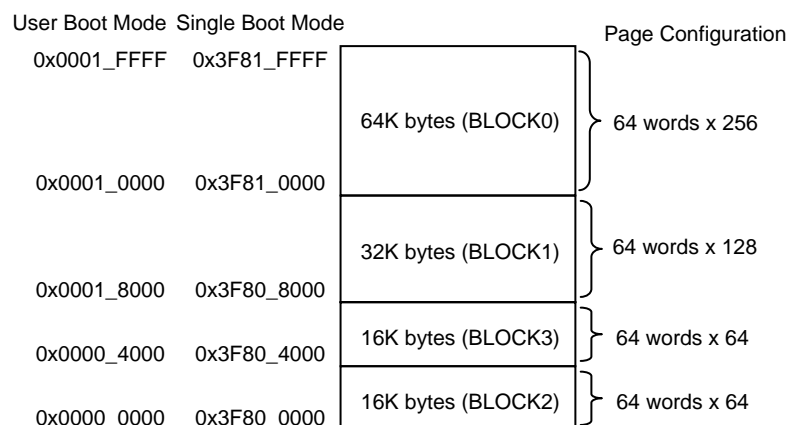| User Boot Mode	Single Boot Mode		Page Configuration
0x0001_FFFF	0x3F81_FFFF	64K bytes (BLOCK0)	64 words x 256
0x0001_0000	0x3F81_0000		
		32K bytes (BLOCK1)	64 words x 128
0x0001_8000	0x3F80_8000		
0x0000_4000	0x3F80_4000	16K bytes (BLOCK3)	64 words x 64
		16K bytes (BLOCK2)	64 words x 64
0x0000_0000	0x3F80_0000		

Fig 20-9 Block Configuration of Flash Memory (TMPM372/3/4FW)

(2) Basic operation

Generally speaking, this flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exceptions other than debug exceptions and reset 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, the operation to apply 32-bit data transmit command to the flash memory is called "bus write cycle." The bus write cycles are to be in a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of a command write operation is in accordance with a predefined specific sequence. If any bus write cycle does not follow a

predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

(Note 1) Command sequences are executed from outside the flash memory area.

(Note 2) Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, don't generate any interrupt (except debug exceptions when a Debugging probe is connected).If such an operation is made, it can result in an unexpected read access to the flash memory and the command sequencer may not be able to correctly recognize the command. While it could cause an abnormal termination of the command sequence, it is also possible that the written command is incorrectly recognized.

(Note 3) For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle that the FLCS RDY/BSY bit 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.

---

(3) Reset

Hardware reset

A hardware reset is used to cancel the operational mode set by the command write operation when forcibly termination during auto programming/ erasing or abnormal termination during auto operations occurs. 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 "Low" level 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 20.2.1 "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.

(4) Commands

    1) Automatic Page Programming

Writing to a flash memory device is to make "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For making "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM372/3/4FW contains 64 words in a page. A 64 word block is defined by a same [31:8] address and it starts from the address [7:0] = 0 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 the FLCS [0] <RDY/BSY> register.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more times irrespective of the data cell value whether it is "1" or "0." Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content can cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. On and after the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at a time). Be sure to use the 32-bit data transfer command in writing commands on and after the fourth bus cycle. In this, any 32-bit data transfer commands shall not be placed across word boundary. On and after the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0." For example, if the top address of a page is not to be written, set the input data of the fourth bus write cycle to 0xFFFFFFFF to command write the data.

Once the fourth bus cycle is executed, it is in the automatic programming operation. This condition can be checked by monitoring the register bit FLCS [0] <RDY/BSY> (See Table 20-13). Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted.

When a single page has been command written normally terminating the automatic page writing process, the FLCS [0] <RDY/BSY> bit is set to "1" and it returns to the read mode. When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FLCS [0] <RDY/BSY> (Table 20-13). If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

> **(Note)** **Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.**

2) Automatic chip erase
   The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

   This condition can be checked by monitoring FLCS [0] <RDY/BSY> (See Table 20-13). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

   Also, any protected blocks cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the mode and will not return to the read mode.

   For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

3) Automatic block erase (fro aech 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 FLCS <RDY/BSY> (See Table 20-13). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary

to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device. It is recommended not to use the device or the identified block anymore.

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 20-18 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 the FLCS <BLPRO> register to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FLCS <RDY/BSY> (See Table 20-13). 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 the FLCS <BLPRO> bits are set to "1" indicating that it is in the protected state (See Table 20-13). 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. The FLCS <RDY/BSY> bit turns to "0" after entering the seventh bus write cycle.**

---

5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on the status of FLCS <BLPRO> whether all the <BLPRO> bits are set to "1" or not if SECBIT<SECBIT> is 0x1. Be sure to check the value of FLCS <BLPRO> before executing the automatic protection bit erase command. See chapter 21 for details.

・**When all the FLCS <BLPRO> bits 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 FLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FLCS will be set to "0x00000001." While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

・**When the FLCS <BLPRO> bits include "0" (not all the protection bits are programmed):**

The protection condition can be canceled by the automatic protection bit erase operation. With this device, protection bits set by an individual block can be erased handling all the blocks at a time as shown in Table 20-19. The target bits are specified in the seventh bus write cycle and when the command is completed, the device is in a condition all the blocks are erased. The protection status of each block can be checked by FLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FLCS <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 FLCS <RDY/BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.**

6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). On and after the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repetitively executed. For returning to the read mode, use the Read/reset command or hardware reset command.

(5) Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the protection status of each block.

Table 20-13 Flash Control Register

FLCS
0x41FF_F020

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read.							

	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	BLPRO3	BLPRO2	BLPRO1	BLPRO0
Read/Write	R				R	R	R	R
After reset	0				(Note 2)	(Note 2)	(Note 2)	(Note 2)
Function	"0" is read.				Protection for Block 3 0: disabled 1:enabled	Protection for Block 2 0: disabled 1:enabled	Protection for Block 1 0: disabled 1:enabled	Protection for Block 0 0: disabled 1:enabled

	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read.							

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	RDY/BSY
Read/Write	R							R
After reset	0							1
Function	"0" is read.							Ready/Busy (Note 1) 0:Auto operating 1:Auto operation terminated

Bit 0: 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."

Bit [19:16]: 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.

(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 microseconds 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.

Table 20-14 Security bit register

SECBIT
0x41FF_F010

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	SECBIT
Read/Write	R							R/W
After reset	0							1
Function	"0" is read							Security bits 0:disabled 1:enabled

(Note) This register is initialized only by power-on reset.

(6) List of Command Sequences

Table 20-15 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle Addr. Data	Second bus cycle Addr. Data	Third bus cycle Addr. Data	Fourth bus cycle Addr. Data	Fifth bus cycle Addr. Data	Sixth bus cycle Addr. Data	Seventh bus cycle Addr. 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 the fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

(Note 1) Always set "0" to the address bits [1:0] in the entire bus cycle. (Recommendable setting values to bits [7:2] are "0".)
(Note 2) 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 data transfer commands. The address [31:16] in each bus write cycle should be the target flash memory address [31:16] of the command sequence. Use "Addr." in the table for the address [15:0].

---

(7)   Address bit configuration for bus write cycles

Table 20-16 Address Bit Configuration for Bus Write Cycles

Address	Addr [31:19]	Addr [18]	Addr [17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10]	Addr [9]	Addr [8]	Addr [7:0]

	Normal bus write cycle address configuration										
Normal commands	Flash area	"0" is recommended.			Command						Addr[1:0]="0" (fixed) Others:0 (recommended)
ID -READ	IA: ID address (Set the fourth bus write cycle address for ID-Read operation)										
	Flash area	"0" is recommended.		ID address		Addr[1:0]="0" (fixed) , Others:0 (recommended)					

	BA: Block address (Set the sixth bus write cycle address for block erase operation)										
Block erase	Block selection (**Table 20-17**)				Addr[1:0]="0" (fixed) , Others:0 (recommended)						
Auto page programming	PA: Program page address (Set the fourth bus write cycle address for page programming operation)										
	Page selection									Addr[1:0]="0" (fixed) Others:0 (recommended)	
Protection bit programming	PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)										
	Flash area	Protection bit selection (**Table 20-18**)		Fixed to "0".				Protection bit selection (**Table 20-18**)		Addr[1:0]="0" (fixed) Others:0 (recommended)	
Protection bit erase	PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)										
	Flash area	Protection bit selection (**Table 20-19**)		"0" is recommended.				Protection bit selection (**Table 20-19**)		Addr[1:0]="0" (fixed) Others:0 (recommended)	
Block erase	BA: Block address (Set the sixth bus write cycle address for block erase operation)										
	Block selection (**Table 20-17**)				Addr[1:0]="0" (fixed) , Others:0 (recommended)						

(Note 1)	Table 20-15 **"Flash Memory Access from the Internal CPU"** can also be used.
(Note 2)	**Address setting can be performed according to the "Normal bus write cycle address configuration" from the first bus cycle.**
(Note 3)	**"0" is recommended" can be changed as necessary.**

Table 20-17 Block Address Table

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
2	0x0000_0000-0x0000_3FFF	0x3F80_0000-0x3F80_3FFF	16
3	0x0000_4000-0x0000_7FFF	0x3F80_4000-0x3F80_7FFF	16
1	0x0000_8000-0x0000_FFFF	0x3F80_8000-0x3F80_FFFF	32
0	0x0001_0000-0x0001_FFFF	0x3F81_0000-0x3F81_FFFF	64

As block address, specify any address in the block to be erased.

(Note) **As for the addresses from the first to the fifth bus cycles, specify the upper 4 bit with the corresponding flash memory addresses of the blocks to be erased.**

Table 20-18 Protection Bit Programming Address Table

Block	Protection bit	The seventh bus write cycle address						
		Address [18]	Address [17]	Address [16]	Address [15:11]	Address [10]	Address [9]	Address [8]
[TMPM372/3/4FW]								
Block0	BLPRO0	0	0	Fixed to "0".			0	0
Block1	BLPRO1	0	0				0	1
Block2	BLPRO2	0	0				1	0
Block3	BLPRO3	0	0				1	1

Table 20-19 Protection Bit Erase Address Table

Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address [18]	Address [17]
Block0~3	BLPRO0~3	0	0

(Note) The protection bit erase command cannot erase by individual block.

Table 20-20 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
00b	0x98	Manufacturer code
01b	0x5A	Device code
10b	Reserved	---
11b	0x11	Macro code

(8) Flowchart
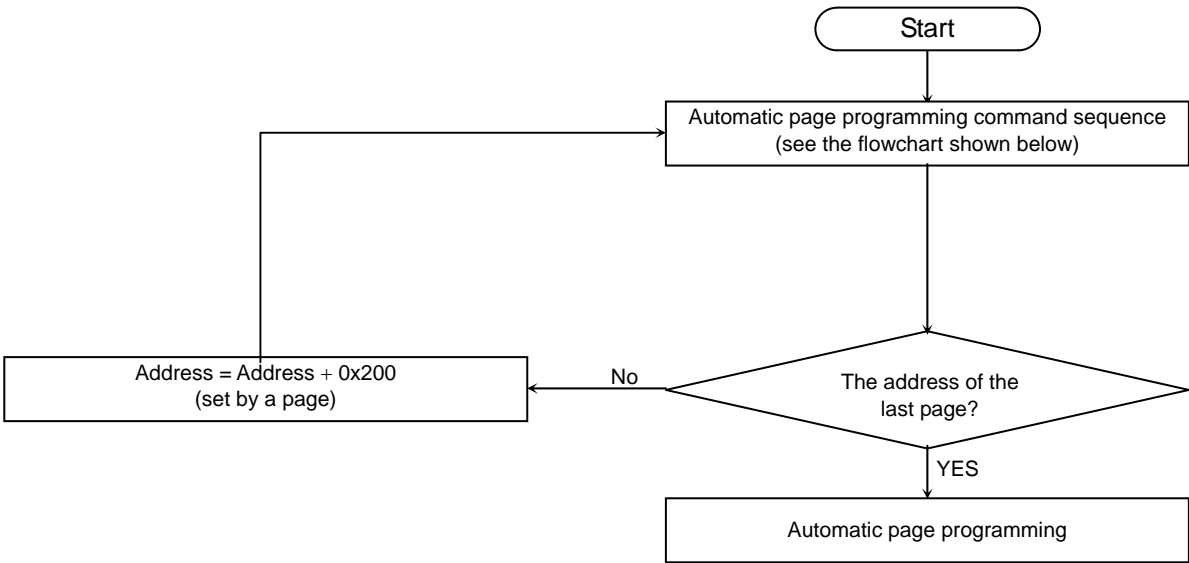
```
                                              ┌──────────────┐
                                              │    Start     │
                                              └──────┬───────┘
                                                     │
                                                     ▼
          ┌─────────────────────────────────────────────────────────────────┐
          │  Automatic page programming command sequence                     │
    ┌────▶│  (see the flowchart shown below)                                  │
    │     └─────────────────────────────────┬─────────────────────────────────┘
    │                                        │
    │                                        ▼
    │                              ╱─────────────────────╲
  ┌─┴──────────────────────┐  No ╱  The address of the    ╲
  │ Address = Address + 0x200│◀───   last page?             
  │ (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)        │
          └────────────────────────────────────────┘
```
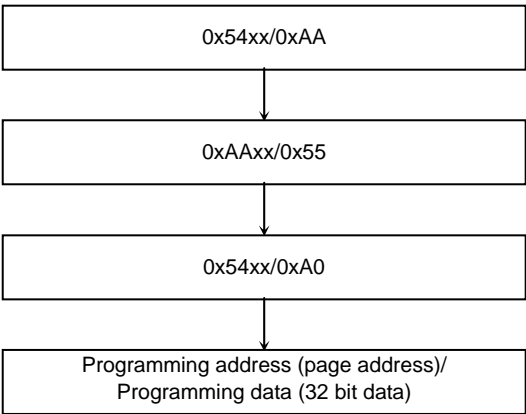
(Note) Command sequence is executed by 0x54xx or 0x55xx.

Fig 20-10 Automatic Programming

```
                              ┌─────────────┐
                              │    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   │
└───────────────────────┘
```
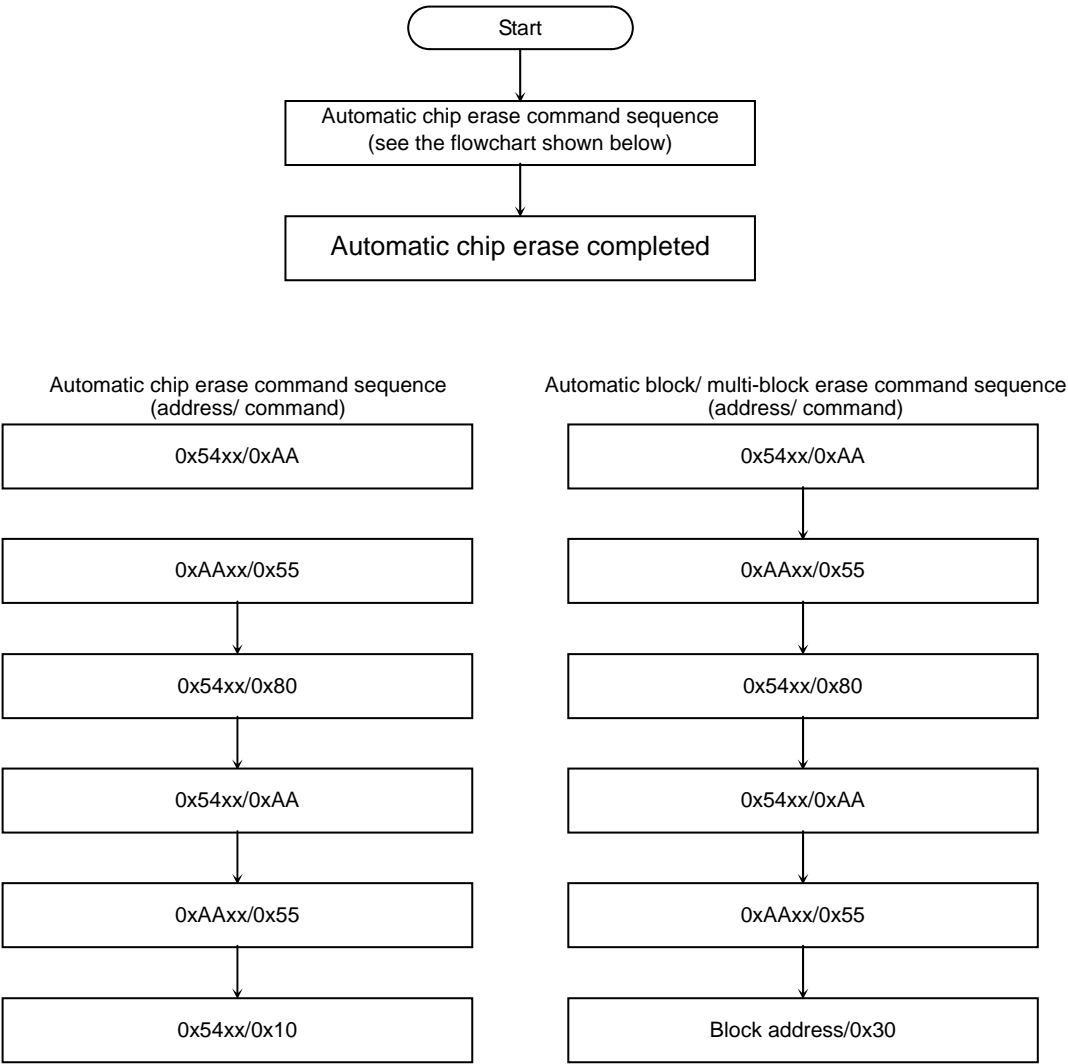
(Note) Command sequence is executed by 0x54xx or 0x55xx.

Fig 20-11 Automatic Erase

21. Protect/security function

21.1 Outline

The TMPM372/3/4FW offers two kinds of ROM protect/ security functions. One is a write/ erase-protect function for the internal flash ROM data. The other is a security function that restricts internal flash ROM data readout and debugging.

21.2 Feature

21.2.1 Internal Flash ROM write/erase protect

The write/ erase-protect function enables the internal flash to prohibit the writing and erasing operation for each block.

This function is available with a single chip mode, single boot mode and writer mode. 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 status of the bits can be monitored by the FCFLCS <BLPRO> bit.

21.2.2 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 to the write/erase-protect function are set to "1".

Note) The FCSECBIT <SECBIT> bit is set to "1" at a power-on reset right after power-on.

Table 21-1 shows details of the restrictions by the security function.

Table 21-1 Restrictions by the security function

Item	Details
1) ROM data readout	The ROM reading operation from CPU is available.
2) Debug port	Communication of JTAG/Serial Wire 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 will erase all the contents of flash memory include protection bits.

21.3 Resisters

The flash control register shows the status of the flash memory operation and the protection of each block.

Table 21-2   Flash control resister

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0							
function	"0" is read							

FCFLCS
0x41FF_F020

	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	BLPRO3	BLPRO2	BLPRO1	BLPRO0
Read/Write	R				R	R	R	R
after Reset	0				(note2)	(note2)	(note2)	(note2)
function	"0" is read				Block3 protect status 0: no protect status. 1: protect status.	Block2 protect status 0: no protect status. 1: protect status.	Block1 protect status 0: no protect status. 1: protect status.	Block0 protect status 0: no protect status. 1: protect status.

	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	"0" is read							

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	RDY/BSY
Read/Write	R							R
after reset	0							1
Function	"0" is read							Ready/ Busy (note1) 0:under auto operation 1:fauto operation is finished

Bit 0: Ready/Busy flag bit
The RDY/BSY output is provided as a means to monitor the status of automatic operation. . 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."

Bit [19:16]: Protection status bits
Each of the protection bits (4 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.

> **(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 microseconds 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 status.

Table 21-3 security bit resister

FCSECBIT
0x41FF_F010

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	SECBIT
Read/Write	R							R/W
after reset	0							1
function	reading data is "0"							security bit 0: disable 1: enable

> **(Note)** This register is initialized only by power-on reset.

21.4 Writing and erasing

21.4.1 Protection bits

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.
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 clearing the FCSECBIT <SECBIT> bit to "0". An attempt to erase protection bits when <SECBIT> bit is "1", it will erases all the contents of flash memory include protection bits. To write and erase the protection bits, command sequence is used.
See chapter "Flash Memory Operation" for details.

21.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on. It can be Rewritten at single chip mode and single boot mode.by the following procedure.

1) Write the code 0xa74a9d23 to FCSECBIT register.
2) Write data within 16 clocks from the writing above.

Note) The above procedure is enabled only when using 32-bit data transfer command.

22 Special Function Registers

[1] Port registers

[2] 16-bit timer (TMRB)

[3] Encoder input (ENC)

[4] Serial interface (UART/SIO)

[5] 12-bit A/D converter (A/DC)

[6] Watchdog timer (WDT)

[7] Clock generator (CG)

[8] Oscillation frequency detector (OFD)

[9] Power on reset (POR), Voltage detecting circuit (VLTD)

[10] Vector engine (VE)

[11] Programmable motor driver (PMD)

[12] Flash

[13] Reserved area

(Note 1)	**As for the internal I/O areas (0x4000_0000~0x4007_FFFF), reading the areas not described in this chapter yields undefined value. Writing these areas is ignored.**
(Note 2)	**<R0> means 0(zero) is read. Writing data is disregarded.**
(Note3)	**Access to the <Reserved> areas is prohibited.**

22.1 Addresses

22.1.1 [1] Port [1/4]

<PORT A>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_0000	PADATA	0x4000_0010		0x4000_0020		0x4000_0030	PAPDN
0x4000_0001	<R0>	0x4000_0011		0x4000_0021		0x4000_0031	<R0>
0x4000_0002	<R0>	0x4000_0012		0x4000_0022		0x4000_0032	<R0>
0x4000_0003	<R0>	0x4000_0013		0x4000_0023		0x4000_0033	<R0>
0x4000_0004	PACR	0x4000_0014		0x4000_0024		0x4000_0034	
0x4000_0005	<R0>	0x4000_0015		0x4000_0025		0x4000_0035	
0x4000_0006	<R0>	0x4000_0016		0x4000_0026		0x4000_0036	
0x4000_0007	<R0>	0x4000_0017		0x4000_0027		0x4000_0037	
0x4000_0008	PAFR1	0x4000_0018		0x4000_0028	PAOD	0x4000_0038	PAIE
0x4000_0009	<R0>	0x4000_0019		0x4000_0029	<R0>	0x4000_0039	<R0>
0x4000_000A	<R0>	0x4000_001A		0x4000_002A	<R0>	0x4000_003A	<R0>
0x4000_000B	<R0>	0x4000_001B		0x4000_002B	<R0>	0x4000_003B	<R0>
0x4000_000C	PAFR2	0x4000_001C		0x4000_002C	PAPUP	0x4000_003C	
0x4000_000D	<R0>	0x4000_001D		0x4000_002D	<R0>	0x4000_003D	
0x4000_000E	<R0>	0x4000_001E		0x4000_002E	<R0>	0x4000_003E	
0x4000_000F	<R0>	0x4000_001F		0x4000_002F	<R0>	0x4000_003F	

<PORT B>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_0040	PBDATA	0x4000_0050		0x4000_0060		0x4000_0070	PBPDN
0x4000_0041	<R0>	0x4000_0051		0x4000_0061		0x4000_0071	<R0>
0x4000_0042	<R0>	0x4000_0052		0x4000_0062		0x4000_0072	<R0>
0x4000_0043	<R0>	0x4000_0053		0x4000_0063		0x4000_0073	<R0>
0x4000_0044	PBCR	0x4000_0054		0x4000_0064		0x4000_0074	
0x4000_0045	<R0>	0x4000_0055		0x4000_0065		0x4000_0075	
0x4000_0046	<R0>	0x4000_0056		0x4000_0066		0x4000_0076	
0x4000_0047	<R0>	0x4000_0057		0x4000_0067		0x4000_0077	
0x4000_0048	PBFR1	0x4000_0058		0x4000_0068	PBOD	0x4000_0078	PBIE
0x4000_0049	<R0>	0x4000_0059		0x4000_0069	<R0>	0x4000_0079	<R0>
0x4000_004A	<R0>	0x4000_005A		0x4000_006A	<R0>	0x4000_007A	<R0>
0x4000_004B	<R0>	0x4000_005B		0x4000_006B	<R0>	0x4000_007B	<R0>
0x4000_004C		0x4000_005C		0x4000_006C	PBPUP	0x4000_007C	
0x4000_004D		0x4000_005D		0x4000_006D	<R0>	0x4000_007D	
0x4000_004E		0x4000_005E		0x4000_006E	<R0>	0x4000_007E	
0x4000_004F		0x4000_005F		0x4000_006F	<R0>	0x4000_007F	

<PORT D>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_00C0	PDDATA	0x4000_00D0		0x4000_00E0		0x4000_00F0	PDPDN
0x4000_00C1	<R0>	0x4000_00D1		0x4000_00E1		0x4000_00F1	<R0>
0x4000_00C2	<R0>	0x4000_00D2		0x4000_00E2		0x4000_00F2	<R0>
0x4000_00C3	<R0>	0x4000_00D3		0x4000_00E3		0x4000_00F3	<R0>
0x4000_00C4	PDCR	0x4000_00D4		0x4000_00E4		0x4000_00F4	
0x4000_00C5	<R0>	0x4000_00D5		0x4000_00E5		0x4000_00F5	
0x4000_00C6	<R0>	0x4000_00D6		0x4000_00E6		0x4000_00F6	
0x4000_00C7	<R0>	0x4000_00D7		0x4000_00E7		0x4000_00F7	
0x4000_00C8	PDFR1	0x4000_00D8		0x4000_00E8	PDOD	0x4000_00F8	PDIE
0x4000_00C9	<R0>	0x4000_00D9		0x4000_00E9	<R0>	0x4000_00F9	<R0>
0x4000_00CA	<R0>	0x4000_00DA		0x4000_00EA	<R0>	0x4000_00FA	<R0>
0x4000_00CB	<R0>	0x4000_00DB		0x4000_00EB	<R0>	0x4000_00FB	<R0>
0x4000_00CC	PDFR2	0x4000_00DC		0x4000_00EC	PDPUP	0x4000_00FC	
0x4000_00CD	<R0>	0x4000_00DD		0x4000_00ED	<R0>	0x4000_00FD	
0x4000_00CE	<R0>	0x4000_00DE		0x4000_00EE	<R0>	0x4000_00FE	
0x4000_00CF	<R0>	0x4000_00DF		0x4000_00EF	<R0>	0x4000_00FF	

Note: Do not access to PD related registers in TMPM374.

TOSHIBA

[1] Port [2/4]
<PORT E>

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4000_0100	PEDATA		0x4000_0110			0x4000_0120			0x4000_0130	PEPDN
0x4000_0101	<R0>		0x4000_0111			0x4000_0121			0x4000_0131	<R0>
0x4000_0102	<R0>		0x4000_0112			0x4000_0122			0x4000_0132	<R0>
0x4000_0103	<R0>		0x4000_0113			0x4000_0123			0x4000_0133	<R0>
0x4000_0104	PECR		0x4000_0114			0x4000_0124			0x4000_0134	
0x4000_0105	<R0>		0x4000_0115			0x4000_0125			0x4000_0135	
0x4000_0106	<R0>		0x4000_0116			0x4000_0126			0x4000_0136	
0x4000_0107	<R0>		0x4000_0117			0x4000_0127			0x4000_0137	
0x4000_0108	PEFR1		0x4000_0118			0x4000_0128	PEOD		0x4000_0138	PEIE
0x4000_0109	<R0>		0x4000_0119			0x4000_0129	<R0>		0x4000_0139	<R0>
0x4000_010A	<R0>		0x4000_011A			0x4000_012A	<R0>		0x4000_013A	<R0>
0x4000_010B	<R0>		0x4000_011B			0x4000_012B	<R0>		0x4000_013B	<R0>
0x4000_010C	PEFR2		0x4000_011C			0x4000_012C	PEPUP		0x4000_013C	
0x4000_010D	<R0>		0x4000_011D			0x4000_012D	<R0>		0x4000_013D	
0x4000_010E	<R0>		0x4000_011E			0x4000_012E	<R0>		0x4000_013E	
0x4000_010F	<R0>		0x4000_011F			0x4000_012F	<R0>		0x4000_013F	

<PORT F>

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4000_0140	PFDATA		0x4000_0150	PFFR3		0x4000_0160			0x4000_0170	PFPDN
0x4000_0141	<R0>		0x4000_0151	<R0>		0x4000_0161			0x4000_0171	<R0>
0x4000_0142	<R0>		0x4000_0152	<R0>		0x4000_0162			0x4000_0172	<R0>
0x4000_0143	<R0>		0x4000_0153	<R0>		0x4000_0163			0x4000_0173	<R0>
0x4000_0144	PFCR		0x4000_0154			0x4000_0164			0x4000_0174	
0x4000_0145	<R0>		0x4000_0155			0x4000_0165			0x4000_0175	
0x4000_0146	<R0>		0x4000_0156			0x4000_0166			0x4000_0176	
0x4000_0147	<R0>		0x4000_0157			0x4000_0167			0x4000_0177	
0x4000_0148	PFFR1		0x4000_0158			0x4000_0168	PFOD		0x4000_0178	PFIE
0x4000_0149	<R0>		0x4000_0159			0x4000_0169	<R0>		0x4000_0179	<R0>
0x4000_014A	<R0>		0x4000_015A			0x4000_016A	<R0>		0x4000_017A	<R0>
0x4000_014B	<R0>		0x4000_015B			0x4000_016B	<R0>		0x4000_017B	<R0>
0x4000_014C	PFFR2		0x4000_015C			0x4000_016C	PFPUP		0x4000_017C	
0x4000_014D	<R0>		0x4000_015D			0x4000_016D	<R0>		0x4000_017D	
0x4000_014E	<R0>		0x4000_015E			0x4000_016E	<R0>		0x4000_017E	
0x4000_014F	<R0>		0x4000_015F			0x4000_016F	<R0>		0x4000_017F	

<PORT G>

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4000_0180	PGDATA		0x4000_0190			0x4000_01A0			0x4000_01B0	PGPDN
0x4000_0181	<R0>		0x4000_0191			0x4000_01A1			0x4000_01B1	<R0>
0x4000_0182	<R0>		0x4000_0192			0x4000_01A2			0x4000_01B2	<R0>
0x4000_0183	<R0>		0x4000_0193			0x4000_01A3			0x4000_01B3	<R0>
0x4000_0184	PGCR		0x4000_0194			0x4000_01A4			0x4000_01B4	
0x4000_0185	<R0>		0x4000_0195			0x4000_01A5			0x4000_01B5	
0x4000_0186	<R0>		0x4000_0196			0x4000_01A6			0x4000_01B6	
0x4000_0187	<R0>		0x4000_0197			0x4000_01A7			0x4000_01B7	
0x4000_0188	PGFR1		0x4000_0198			0x4000_01A8	PGOD		0x4000_01B8	PGIE
0x4000_0189	<R0>		0x4000_0199			0x4000_01A9	<R0>		0x4000_01B9	<R0>
0x4000_018A	<R0>		0x4000_019A			0x4000_01AA	<R0>		0x4000_01BA	<R0>
0x4000_018B	<R0>		0x4000_019B			0x4000_01AB	<R0>		0x4000_01BB	<R0>
0x4000_018C			0x4000_019C			0x4000_01AC	PGPUP		0x4000_01BC	
0x4000_018D			0x4000_019D			0x4000_01AD	<R0>		0x4000_01BD	
0x4000_018E			0x4000_019E			0x4000_01AE	<R0>		0x4000_01BE	
0x4000_018F			0x4000_019F			0x4000_01AF	<R0>		0x4000_01BF	

[1] Port [3/4]

<PORT I>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_0200	PIDATA	0x4000_0210		0x4000_0220		0x4000_0230	PIPDN
0x4000_0201	<R0>	0x4000_0211		0x4000_0221		0x4000_0231	<R0>
0x4000_0202	<R0>	0x4000_0212		0x4000_0222		0x4000_0232	<R0>
0x4000_0203	<R0>	0x4000_0213		0x4000_0223		0x4000_0233	<R0>
0x4000_0204	PICR	0x4000_0214		0x4000_0224		0x4000_0234	
0x4000_0205	<R0>	0x4000_0215		0x4000_0225		0x4000_0235	
0x4000_0206	<R0>	0x4000_0216		0x4000_0226		0x4000_0236	
0x4000_0207	<R0>	0x4000_0217		0x4000_0227		0x4000_0237	
0x4000_0208		0x4000_0218		0x4000_0228	PIOD	0x4000_0238	PIIE
0x4000_0209		0x4000_0219		0x4000_0229	<R0>	0x4000_0239	<R0>
0x4000_020A		0x4000_021A		0x4000_022A	<R0>	0x4000_023A	<R0>
0x4000_020B		0x4000_021B		0x4000_022B	<R0>	0x4000_023B	<R0>
0x4000_020C		0x4000_021C		0x4000_022C	PIPUP	0x4000_023C	
0x4000_020D		0x4000_021D		0x4000_022D	<R0>	0x4000_023D	
0x4000_020E		0x4000_021E		0x4000_022E	<R0>	0x4000_023E	
0x4000_020F		0x4000_021F		0x4000_022F	<R0>	0x4000_023F	

<PORT J>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_0240	PJDATA	0x4000_0250		0x4000_0260		0x4000_0270	PJPDN
0x4000_0241	<R0>	0x4000_0251		0x4000_0261		0x4000_0271	<R0>
0x4000_0242	<R0>	0x4000_0252		0x4000_0262		0x4000_0272	<R0>
0x4000_0243	<R0>	0x4000_0253		0x4000_0263		0x4000_0273	<R0>
0x4000_0244	PJCR	0x4000_0254		0x4000_0264		0x4000_0274	
0x4000_0245	<R0>	0x4000_0255		0x4000_0265		0x4000_0275	
0x4000_0246	<R0>	0x4000_0256		0x4000_0266		0x4000_0276	
0x4000_0247	<R0>	0x4000_0257		0x4000_0267		0x4000_0277	
0x4000_0248	PJFR1	0x4000_0258		0x4000_0268	PJOD	0x4000_0278	PJIE
0x4000_0249	<R0>	0x4000_0259		0x4000_0269	<R0>	0x4000_0279	<R0>
0x4000_024A	<R0>	0x4000_025A		0x4000_026A	<R0>	0x4000_027A	<R0>
0x4000_024B	<R0>	0x4000_025B		0x4000_026B	<R0>	0x4000_027B	<R0>
0x4000_024C		0x4000_025C		0x4000_026C	PJPUP	0x4000_027C	
0x4000_024D		0x4000_025D		0x4000_026D	<R0>	0x4000_027D	
0x4000_024E		0x4000_025E		0x4000_026E	<R0>	0x4000_027E	
0x4000_024F		0x4000_025F		0x4000_026F	<R0>	0x4000_027F	

<PORT K>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_0280	PKDATA	0x4000_0290		0x4000_02A0		0x4000_02B0	PKPDN
0x4000_0281	<R0>	0x4000_0291		0x4000_02A1		0x4000_02B1	<R0>
0x4000_0282	<R0>	0x4000_0292		0x4000_02A2		0x4000_02B2	<R0>
0x4000_0283	<R0>	0x4000_0293		0x4000_02A3		0x4000_02B3	<R0>
0x4000_0284	PKCR	0x4000_0294		0x4000_02A4		0x4000_02B4	
0x4000_0285	<R0>	0x4000_0295		0x4000_02A5		0x4000_02B5	
0x4000_0286	<R0>	0x4000_0296		0x4000_02A6		0x4000_02B6	
0x4000_0287	<R0>	0x4000_0297		0x4000_02A7		0x4000_02B7	
0x4000_0288	PKFR1	0x4000_0298		0x4000_02A8	PKOD	0x4000_02B8	PKIE
0x4000_0289	<R0>	0x4000_0299		0x4000_02A9	<R0>	0x4000_02B9	<R0>
0x4000_028A	<R0>	0x4000_029A		0x4000_02AA	<R0>	0x4000_02BA	<R0>
0x4000_028B	<R0>	0x4000_029B		0x4000_02AB	<R0>	0x4000_02BB	<R0>
0x4000_028C		0x4000_029C		0x4000_02AC	PKPUP	0x4000_02BC	
0x4000_028D		0x4000_029D		0x4000_02AD	<R0>	0x4000_02BD	
0x4000_028E		0x4000_029E		0x4000_02AE	<R0>	0x4000_02BE	
0x4000_028F		0x4000_029F		0x4000_02AF	<R0>	0x4000_02BF	

[1] Port [4/4]
<PORT M>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4000_0300	PMDATA	0x4000_0310		0x4000_0320		0x4000_0330	PMPDN
0x4000_0301	⟨R0⟩	0x4000_0311		0x4000_0321		0x4000_0331	⟨R0⟩
0x4000_0302	⟨R0⟩	0x4000_0312		0x4000_0322		0x4000_0332	⟨R0⟩
0x4000_0303	⟨R0⟩	0x4000_0313		0x4000_0323		0x4000_0333	⟨R0⟩
0x4000_0304	PMCR	0x4000_0314		0x4000_0324		0x4000_0334	
0x4000_0305	⟨R0⟩	0x4000_0315		0x4000_0325		0x4000_0335	
0x4000_0306	⟨R0⟩	0x4000_0316		0x4000_0326		0x4000_0336	
0x4000_0307	⟨R0⟩	0x4000_0317		0x4000_0327		0x4000_0337	
0x4000_0308		0x4000_0318		0x4000_0328	PMOD	0x4000_0338	PMIE
0x4000_0309		0x4000_0319		0x4000_0329	⟨R0⟩	0x4000_0339	⟨R0⟩
0x4000_030A		0x4000_031A		0x4000_032A	⟨R0⟩	0x4000_033A	⟨R0⟩
0x4000_030B		0x4000_031B		0x4000_032B	⟨R0⟩	0x4000_033B	⟨R0⟩
0x4000_030C		0x4000_031C		0x4000_032C	PMPUP	0x4000_033C	
0x4000_030D		0x4000_031D		0x4000_032D	⟨R0⟩	0x4000_033D	
0x4000_030E		0x4000_031E		0x4000_032E	⟨R0⟩	0x4000_033E	
0x4000_030F		0x4000_031F		0x4000_032F	⟨R0⟩	0x4000_033F	

22.1.2 [2] 16-bit timer [1/3]

\<TMRB0\>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0000	TB0EN	0x4001_0010	TB0FFCR	0x4001_0020	TB0RG0	0x4001_0030	
0x4001_0001	\<R0\>	0x4001_0011	\<R0\>	0x4001_0021	"	0x4001_0031	
0x4001_0002	\<R0\>	0x4001_0012	\<R0\>	0x4001_0022	\<R0\>	0x4001_0032	
0x4001_0003	\<R0\>	0x4001_0013	\<R0\>	0x4001_0023	\<R0\>	0x4001_0033	
0x4001_0004	TB0RUN	0x4001_0014	TB0ST	0x4001_0024	TB0RG1	0x4001_0034	
0x4001_0005	\<R0\>	0x4001_0015	\<R0\>	0x4001_0025	"	0x4001_0035	
0x4001_0006	\<R0\>	0x4001_0016	\<R0\>	0x4001_0026	\<R0\>	0x4001_0036	
0x4001_0007	\<R0\>	0x4001_0017	\<R0\>	0x4001_0027	\<R0\>	0x4001_0037	
0x4001_0008	TB0CR	0x4001_0018	TB0IM	0x4001_0028	TB0CP0	0x4001_0038	
0x4001_0009	\<R0\>	0x4001_0019	\<R0\>	0x4001_0029	"	0x4001_0039	
0x4001_000A	\<R0\>	0x4001_001A	\<R0\>	0x4001_002A	\<R0\>	0x4001_003A	
0x4001_000B	\<R0\>	0x4001_001B	\<R0\>	0x4001_002B	\<R0\>	0x4001_003B	
0x4001_000C	TB0MOD	0x4001_001C	TB0UC	0x4001_002C	TB0CP1	0x4001_003C	
0x4001_000D	\<R0\>	0x4001_001D	"	0x4001_002D	"	0x4001_003D	
0x4001_000E	\<R0\>	0x4001_001E	\<R0\>	0x4001_002E	\<R0\>	0x4001_003E	
0x4001_000F	\<R0\>	0x4001_001F	\<R0\>	0x4001_002F	\<R0\>	0x4001_003F	

\<TMRB1\>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0040	TB1EN	0x4001_0050	TB1FFCR	0x4001_0060	TB1RG0	0x4001_0070	
0x4001_0041	\<R0\>	0x4001_0051	\<R0\>	0x4001_0061	"	0x4001_0071	
0x4001_0042	\<R0\>	0x4001_0052	\<R0\>	0x4001_0062	\<R0\>	0x4001_0072	
0x4001_0043	\<R0\>	0x4001_0053	\<R0\>	0x4001_0063	\<R0\>	0x4001_0073	
0x4001_0044	TB1RUN	0x4001_0054	TB1ST	0x4001_0064	TB1RG1	0x4001_0074	
0x4001_0045	\<R0\>	0x4001_0055	\<R0\>	0x4001_0065	"	0x4001_0075	
0x4001_0046	\<R0\>	0x4001_0056	\<R0\>	0x4001_0066	\<R0\>	0x4001_0076	
0x4001_0047	\<R0\>	0x4001_0057	\<R0\>	0x4001_0067	\<R0\>	0x4001_0077	
0x4001_0048	TB1CR	0x4001_0058	TB1IM	0x4001_0068	TB1CP0	0x4001_0078	
0x4001_0049	\<R0\>	0x4001_0059	\<R0\>	0x4001_0069	"	0x4001_0079	
0x4001_004A	\<R0\>	0x4001_005A	\<R0\>	0x4001_006A	\<R0\>	0x4001_007A	
0x4001_004B	\<R0\>	0x4001_005B	\<R0\>	0x4001_006B	\<R0\>	0x4001_007B	
0x4001_004C	TB1MOD	0x4001_005C	TB1UC	0x4001_006C	TB1CP1	0x4001_007C	
0x4001_004D	\<R0\>	0x4001_005D	"	0x4001_006D	"	0x4001_007D	
0x4001_004E	\<R0\>	0x4001_005E	\<R0\>	0x4001_006E	\<R0\>	0x4001_007E	
0x4001_004F	\<R0\>	0x4001_005F	\<R0\>	0x4001_006F	\<R0\>	0x4001_007F	

\<TMRB2\>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0080	TB2EN	0x4001_0090	TB2FFCR	0x4001_00A0	TB2RG0	0x4001_00B0	
0x4001_0081	\<R0\>	0x4001_0091	\<R0\>	0x4001_00A1	"	0x4001_00B1	
0x4001_0082	\<R0\>	0x4001_0092	\<R0\>	0x4001_00A2	\<R0\>	0x4001_00B2	
0x4001_0083	\<R0\>	0x4001_0093	\<R0\>	0x4001_00A3	\<R0\>	0x4001_00B3	
0x4001_0084	TB2RUN	0x4001_0094	TB2ST	0x4001_00A4	TB2RG1	0x4001_00B4	
0x4001_0085	\<R0\>	0x4001_0095	\<R0\>	0x4001_00A5	"	0x4001_00B5	
0x4001_0086	\<R0\>	0x4001_0096	\<R0\>	0x4001_00A6	\<R0\>	0x4001_00B6	
0x4001_0087	\<R0\>	0x4001_0097	\<R0\>	0x4001_00A7	\<R0\>	0x4001_00B7	
0x4001_0088	TB2CR	0x4001_0098	TB2IM	0x4001_00A8	TB2CP0	0x4001_00B8	
0x4001_0089	\<R0\>	0x4001_0099	\<R0\>	0x4001_00A9	"	0x4001_00B9	
0x4001_008A	\<R0\>	0x4001_009A	\<R0\>	0x4001_00AA	\<R0\>	0x4001_00BA	
0x4001_008B	\<R0\>	0x4001_009B	\<R0\>	0x4001_00AB	\<R0\>	0x4001_00BB	
0x4001_008C	TB2MOD	0x4001_009C	TB2UC	0x4001_00AC	TB2CP1	0x4001_00BC	
0x4001_008D	\<R0\>	0x4001_009D	"	0x4001_00AD	"	0x4001_00BD	
0x4001_008E	\<R0\>	0x4001_009E	\<R0\>	0x4001_00AE	\<R0\>	0x4001_00BE	
0x4001_008F	\<R0\>	0x4001_009F	\<R0\>	0x4001_00AF	\<R0\>	0x4001_00BF	

Note: Do not access to TB0FFCR, TB1FFCR, TB2FFCR in TMPM373/374.

[2] 16-bit timer [2/3]

<TMRB3>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_00C0	TB3EN	0x4001_00D0	TB3FFCR	0x4001_00E0	TB3RG0	0x4001_00F0	
0x4001_00C1	<R0>	0x4001_00D1	<R0>	0x4001_00E1	"	0x4001_00F1	
0x4001_00C2	<R0>	0x4001_00D2	<R0>	0x4001_00E2	<R0>	0x4001_00F2	
0x4001_00C3	<R0>	0x4001_00D3	<R0>	0x4001_00E3	<R0>	0x4001_00F3	
0x4001_00C4	TB3RUN	0x4001_00D4	TB3ST	0x4001_00E4	TB3RG1	0x4001_00F4	
0x4001_00C5	<R0>	0x4001_00D5	<R0>	0x4001_00E5	"	0x4001_00F5	
0x4001_00C6	<R0>	0x4001_00D6	<R0>	0x4001_00E6	<R0>	0x4001_00F6	
0x4001_00C7	<R0>	0x4001_00D7	<R0>	0x4001_00E7	<R0>	0x4001_00F7	
0x4001_00C8	TB3CR	0x4001_00D8	TB3IM	0x4001_00E8	TB3CP0	0x4001_00F8	
0x4001_00C9	<R0>	0x4001_00D9	<R0>	0x4001_00E9	"	0x4001_00F9	
0x4001_00CA	<R0>	0x4001_00DA	<R0>	0x4001_00EA	<R0>	0x4001_00FA	
0x4001_00CB	<R0>	0x4001_00DB	<R0>	0x4001_00EB	<R0>	0x4001_00FB	
0x4001_00CC	TB3MOD	0x4001_00DC	TB3UC	0x4001_00EC	TB3CP1	0x4001_00FC	
0x4001_00CD	<R0>	0x4001_00DD	"	0x4001_00ED	"	0x4001_00FD	
0x4001_00CE	<R0>	0x4001_00DE	<R0>	0x4001_00EE	<R0>	0x4001_00FE	
0x4001_00CF	<R0>	0x4001_00DF	<R0>	0x4001_00EF	<R0>	0x4001_00FF	

<TMRB4>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0100	TB4EN	0x4001_0110	TB4FFCR	0x4001_0120	TB4RG0	0x4001_0130	
0x4001_0101	<R0>	0x4001_0111	<R0>	0x4001_0121	"	0x4001_0131	
0x4001_0102	<R0>	0x4001_0112	<R0>	0x4001_0122	<R0>	0x4001_0132	
0x4001_0103	<R0>	0x4001_0113	<R0>	0x4001_0123	<R0>	0x4001_0133	
0x4001_0104	TB4RUN	0x4001_0114	TB4ST	0x4001_0124	TB4RG1	0x4001_0134	
0x4001_0105	<R0>	0x4001_0115	<R0>	0x4001_0125	"	0x4001_0135	
0x4001_0106	<R0>	0x4001_0116	<R0>	0x4001_0126	<R0>	0x4001_0136	
0x4001_0107	<R0>	0x4001_0117	<R0>	0x4001_0127	<R0>	0x4001_0137	
0x4001_0108	TB4CR	0x4001_0118	TB4IM	0x4001_0128	TB4CP0	0x4001_0138	
0x4001_0109	<R0>	0x4001_0119	<R0>	0x4001_0129	"	0x4001_0139	
0x4001_010A	<R0>	0x4001_011A	<R0>	0x4001_012A	<R0>	0x4001_013A	
0x4001_010B	<R0>	0x4001_011B	<R0>	0x4001_012B	<R0>	0x4001_013B	
0x4001_010C	TB4MOD	0x4001_011C	TB4UC	0x4001_012C	TB4CP1	0x4001_013C	
0x4001_010D	<R0>	0x4001_011D	"	0x4001_012D	"	0x4001_013D	"
0x4001_010E	<R0>	0x4001_011E	<R0>	0x4001_012E	<R0>	0x4001_013E	
0x4001_010F	<R0>	0x4001_011F	<R0>	0x4001_012F	<R0>	0x4001_013F	

<TMRB5>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0140	TB5EN	0x4001_0150	Reserved	0x4001_0160	TB5RG0	0x4001_0170	
0x4001_0141	<R0>	0x4001_0151	Reserved	0x4001_0161	"	0x4001_0171	
0x4001_0142	<R0>	0x4001_0152	Reserved	0x4001_0162	<R0>	0x4001_0172	
0x4001_0143	<R0>	0x4001_0153	Reserved	0x4001_0163	<R0>	0x4001_0173	
0x4001_0144	TB5RUN	0x4001_0154	TB5ST	0x4001_0164	TB5RG1	0x4001_0174	
0x4001_0145	<R0>	0x4001_0155	<R0>	0x4001_0165	"	0x4001_0175	
0x4001_0146	<R0>	0x4001_0156	<R0>	0x4001_0166	<R0>	0x4001_0176	
0x4001_0147	<R0>	0x4001_0157	<R0>	0x4001_0167	<R0>	0x4001_0177	
0x4001_0148	TB5CR	0x4001_0158	TB5IM	0x4001_0168	TB5CP0	0x4001_0178	
0x4001_0149	<R0>	0x4001_0159	<R0>	0x4001_0169	"	0x4001_0179	
0x4001_014A	<R0>	0x4001_015A	<R0>	0x4001_016A	<R0>	0x4001_017A	
0x4001_014B	<R0>	0x4001_015B	<R0>	0x4001_016B	<R0>	0x4001_017B	
0x4001_014C	TB5MOD	0x4001_015C	TB5UC	0x4001_016C	TB5CP1	0x4001_017C	
0x4001_014D	<R0>	0x4001_015D	"	0x4001_016D	"	0x4001_017D	
0x4001_014E	<R0>	0x4001_015E	<R0>	0x4001_016E	<R0>	0x4001_017E	
0x4001_014F	<R0>	0x4001_015F	<R0>	0x4001_016F	<R0>	0x4001_017F	

Note: Do not access to TB4FFCR TMPM373/374.

[2] 16-bit timer [3/3]

<TMRB6>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0180	TB6EN	0x4001_0190	TB6FFCR	0x4001_01A0	TB6RG0	0x4001_01B0	
0x4001_0181	<R0>	0x4001_0191	<R0>	0x4001_01A1	"	0x4001_01B1	
0x4001_0182	<R0>	0x4001_0192	<R0>	0x4001_01A2	<R0>	0x4001_01B2	
0x4001_0183	<R0>	0x4001_0193	<R0>	0x4001_01A3	<R0>	0x4001_01B3	
0x4001_0184	TB6RUN	0x4001_0194	TB6ST	0x4001_01A4	TB6RG1	0x4001_01B4	
0x4001_0185	<R0>	0x4001_0195	<R0>	0x4001_01A5	"	0x4001_01B5	
0x4001_0186	<R0>	0x4001_0196	<R0>	0x4001_01A6	<R0>	0x4001_01B6	
0x4001_0187	<R0>	0x4001_0197	<R0>	0x4001_01A7	<R0>	0x4001_01B7	
0x4001_0188	TB6CR	0x4001_0198	TB6IM	0x4001_01A8	TB6CP0	0x4001_01B8	
0x4001_0189	<R0>	0x4001_0199	<R0>	0x4001_01A9	"	0x4001_01B9	
0x4001_018A	<R0>	0x4001_019A	<R0>	0x4001_01AA	<R0>	0x4001_01BA	
0x4001_018B	<R0>	0x4001_019B	<R0>	0x4001_01AB	<R0>	0x4001_01BB	
0x4001_018C	TB6MOD	0x4001_019C	TB6UC	0x4001_01AC	TB6CP1	0x4001_01BC	
0x4001_018D	<R0>	0x4001_019D	"	0x4001_01AD	"	0x4001_01BD	
0x4001_018E	<R0>	0x4001_019E	<R0>	0x4001_01AE	<R0>	0x4001_01BE	
0x4001_018F	<R0>	0x4001_019F	<R0>	0x4001_01AF	<R0>	0x4001_01BF	

<TMRB7>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_01C0	TB7EN	0x4001_01D0	TB7FFCR	0x4001_01E0	TB7RG0	0x4001_01F0	
0x4001_01C1	<R0>	0x4001_01D1	<R0>	0x4001_01E1	"	0x4001_01F1	
0x4001_01C2	<R0>	0x4001_01D2	<R0>	0x4001_01E2	<R0>	0x4001_01F2	
0x4001_01C3	<R0>	0x4001_01D3	<R0>	0x4001_01E3	<R0>	0x4001_01F3	
0x4001_01C4	TB7RUN	0x4001_01D4	TB7ST	0x4001_01E4	TB7RG1	0x4001_01F4	
0x4001_01C5	<R0>	0x4001_01D5	<R0>	0x4001_01E5	"	0x4001_01F5	
0x4001_01C6	<R0>	0x4001_01D6	<R0>	0x4001_01E6	<R0>	0x4001_01F6	
0x4001_01C7	<R0>	0x4001_01D7	<R0>	0x4001_01E7	<R0>	0x4001_01F7	
0x4001_01C8	TB7CR	0x4001_01D8	TB7IM	0x4001_01E8	TB7CP0	0x4001_01F8	
0x4001_01C9	<R0>	0x4001_01D9	<R0>	0x4001_01E9	"	0x4001_01F9	
0x4001_01CA	<R0>	0x4001_01DA	<R0>	0x4001_01EA	<R0>	0x4001_01FA	
0x4001_01CB	<R0>	0x4001_01DB	<R0>	0x4001_01EB	<R0>	0x4001_01FB	
0x4001_01CC	TB7MOD	0x4001_01DC	TB7UC	0x4001_01EC	TB7CP1	0x4001_01FC	
0x4001_01CD	<R0>	0x4001_01DD	"	0x4001_01ED	"	0x4001_01FD	
0x4001_01CE	<R0>	0x4001_01DE	<R0>	0x4001_01EE	<R0>	0x4001_01FE	
0x4001_01CF	<R0>	0x4001_01DF	<R0>	0x4001_01EF	<R0>	0x4001_01FF	

Note: Do not access to TB7FFCR in TMPM374.

22.1.3 [3] Encoder input (ENC)

<ENC1>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4001_0500	EN1TNCR	0x4001_0510		0x4001_0520		0x4001_0530	
0x4001_0501	"	0x4001_0511		0x4001_0521		0x4001_0531	
0x4001_0502	"	0x4001_0512		0x4001_0522		0x4001_0532	
0x4001_0503	<R0>	0x4001_0513		0x4001_0523		0x4001_0533	
0x4001_0504	EN1RELOAD	0x4001_0514		0x4001_0524		0x4001_0534	
0x4001_0505	"	0x4001_0515		0x4001_0525		0x4001_0535	
0x4001_0506	<R0>	0x4001_0516		0x4001_0526		0x4001_0536	
0x4001_0507	<R0>	0x4001_0517		0x4001_0527		0x4001_0537	
0x4001_0508	EN1INT	0x4001_0518		0x4001_0528		0x4001_0538	
0x4001_0509	"	0x4001_0519		0x4001_0529		0x4001_0539	
0x4001_050A	"	0x4001_051A		0x4001_052A		0x4001_053A	
0x4001_050B	<R0>	0x4001_051B		0x4001_052B		0x4001_053B	
0x4001_050C	EN1CNT	0x4001_051C		0x4001_052C		0x4001_053C	
0x4001_050D	"	0x4001_051D		0x4001_052D		0x4001_053D	
0x4001_050E	"	0x4001_051E		0x4001_052E		0x4001_053E	
0x4001_050F	<R0>	0x4001_051F		0x4001_052F		0x4001_053F	

22.1.4　[4] Serial interface (UART/SIO) [1/2]

<SIO0>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4002_0080	SC0EN	0x4002_0090	SC0BRCR	0x4002_00A0	SC0RFC	0x4002_00B0	SC0FCNF
0x4002_0081	<R0>	0x4002_0091	<R0>	0x4002_00A1	<R0>	0x4002_00B1	<R0>
0x4002_0082	<R0>	0x4002_0092	<R0>	0x4002_00A2	<R0>	0x4002_00B2	<R0>
0x4002_0083	<R0>	0x4002_0093	<R0>	0x4002_00A3	<R0>	0x4002_00B3	<R0>
0x4002_0084	SC0BUF	0x4002_0094	SC0BRADD	0x4002_00A4	SC0TFC	0x4002_00B4	
0x4002_0085	<R0>	0x4002_0095	<R0>	0x4002_00A5	<R0>	0x4002_00B5	
0x4002_0086	<R0>	0x4002_0096	<R0>	0x4002_00A6	<R0>	0x4002_00B6	
0x4002_0087	<R0>	0x4002_0097	<R0>	0x4002_00A7	<R0>	0x4002_00B7	
0x4002_0088	SC0CR	0x4002_0098	SC0MOD1	0x4002_00A8	SC0RST	0x4002_00B8	
0x4002_0089	<R0>	0x4002_0099	<R0>	0x4002_00A9	<R0>	0x4002_00B9	
0x4002_008A	<R0>	0x4002_009A	<R0>	0x4002_00AA	<R0>	0x4002_00BA	
0x4002_008B	<R0>	0x4002_009B	<R0>	0x4002_00AB	<R0>	0x4002_00BB	
0x4002_008C	SC0MOD0	0x4002_009C	SC0MOD2	0x4002_00AC	SC0TST	0x4002_00BC	
0x4002_008D	<R0>	0x4002_009D	<R0>	0x4002_00AD	<R0>	0x4002_00BD	
0x4002_008E	<R0>	0x4002_009E	<R0>	0x4002_00AE	<R0>	0x4002_00BE	
0x4002_008F	<R0>	0x4002_009F	<R0>	0x4002_00AF	<R0>	0x4002_00BF	

<SIO1>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4002_00C0	SC1EN	0x4002_00D0	SC1BRCR	0x4002_00E0	SC1RFC	0x4002_00F0	SC1FCNF
0x4002_00C1	<R0>	0x4002_00D1	<R0>	0x4002_00E1	<R0>	0x4002_00F1	<R0>
0x4002_00C2	<R0>	0x4002_00D2	<R0>	0x4002_00E2	<R0>	0x4002_00F2	<R0>
0x4002_00C3	<R0>	0x4002_00D3	<R0>	0x4002_00E3	<R0>	0x4002_00F3	<R0>
0x4002_00C4	SC1BUF	0x4002_00D4	SC1BRADD	0x4002_00E4	SC1TFC	0x4002_00F4	
0x4002_00C5	<R0>	0x4002_00D5	<R0>	0x4002_00E5	<R0>	0x4002_00F5	
0x4002_00C6	<R0>	0x4002_00D6	<R0>	0x4002_00E6	<R0>	0x4002_00F6	
0x4002_00C7	<R0>	0x4002_00D7	<R0>	0x4002_00E7	<R0>	0x4002_00F7	
0x4002_00C8	SC1CR	0x4002_00D8	SC1MOD1	0x4002_00E8	SC1RST	0x4002_00F8	
0x4002_00C9	<R0>	0x4002_00D9	<R0>	0x4002_00E9	<R0>	0x4002_00F9	
0x4002_00CA	<R0>	0x4002_00DA	<R0>	0x4002_00EA	<R0>	0x4002_00FA	
0x4002_00CB	<R0>	0x4002_00DB	<R0>	0x4002_00EB	<R0>	0x4002_00FB	
0x4002_00CC	SC1MOD0	0x4002_00DC	SC1MOD2	0x4002_00EC	SC1TST	0x4002_00FC	
0x4002_00CD	<R0>	0x4002_00DD	<R0>	0x4002_00ED	<R0>	0x4002_00FD	
0x4002_00CE	<R0>	0x4002_00DE	<R0>	0x4002_00EE	<R0>	0x4002_00FE	
0x4002_00CF	<R0>	0x4002_00DF	<R0>	0x4002_00EF	<R0>	0x4002_00FF	

<SIO2>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4002_0100	SC2EN	0x4002_0110	SC2BRCR	0x4002_0120	SC2RFC	0x4002_0130	SC2FCNF
0x4002_0101	<R0>	0x4002_0111	<R0>	0x4002_0121	<R0>	0x4002_0131	<R0>
0x4002_0102	<R0>	0x4002_0112	<R0>	0x4002_0122	<R0>	0x4002_0132	<R0>
0x4002_0103	<R0>	0x4002_0113	<R0>	0x4002_0123	<R0>	0x4002_0133	<R0>
0x4002_0104	SC2BUF	0x4002_0114	SC2BRADD	0x4002_0124	SC2TFC	0x4002_0134	
0x4002_0105	<R0>	0x4002_0115	<R0>	0x4002_0125	<R0>	0x4002_0135	
0x4002_0106	<R0>	0x4002_0116	<R0>	0x4002_0126	<R0>	0x4002_0136	
0x4002_0107	<R0>	0x4002_0117	<R0>	0x4002_0127	<R0>	0x4002_0137	
0x4002_0108	SC2CR	0x4002_0118	SC2MOD1	0x4002_0128	SC2RST	0x4002_0138	
0x4002_0109	<R0>	0x4002_0119	<R0>	0x4002_0129	<R0>	0x4002_0139	
0x4002_010A	<R0>	0x4002_011A	<R0>	0x4002_012A	<R0>	0x4002_013A	
0x4002_010B	<R0>	0x4002_011B	<R0>	0x4002_012B	<R0>	0x4002_013B	
0x4002_010C	SC2MOD0	0x4002_011C	SC2MOD2	0x4002_012C	SC2TST	0x4002_013C	
0x4002_010D	<R0>	0x4002_011D	<R0>	0x4002_012D	<R0>	0x4002_013D	
0x4002_010E	<R0>	0x4002_011E	<R0>	0x4002_012E	<R0>	0x4002_013E	
0x4002_010F	<R0>	0x4002_011F	<R0>	0x4002_012F	<R0>	0x4002_013F	

[4] Serial interface (UART/SIO) [2/2]
<SIO3>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4002_0140	SC3EN	0x4002_0150	SC3BRCR	0x4002_0160	SC3RFC	0x4002_0170	SC3FCNF
0x4002_0141	<R0>	0x4002_0151	<R0>	0x4002_0161	<R0>	0x4002_0171	<R0>
0x4002_0142	<R0>	0x4002_0152	<R0>	0x4002_0162	<R0>	0x4002_0172	<R0>
0x4002_0143	<R0>	0x4002_0153	<R0>	0x4002_0163	<R0>	0x4002_0173	<R0>
0x4002_0144	SC3BUF	0x4002_0154	SC3BRADD	0x4002_0164	SC3TFC	0x4002_0174	
0x4002_0145	<R0>	0x4002_0155	<R0>	0x4002_0165	<R0>	0x4002_0175	
0x4002_0146	<R0>	0x4002_0156	<R0>	0x4002_0166	<R0>	0x4002_0176	
0x4002_0147	<R0>	0x4002_0157	<R0>	0x4002_0167	<R0>	0x4002_0177	
0x4002_0148	SC3CR	0x4002_0158	SC3MOD1	0x4002_0168	SC3RST	0x4002_0178	
0x4002_0149	<R0>	0x4002_0159	<R0>	0x4002_0169	<R0>	0x4002_0179	
0x4002_014A	<R0>	0x4002_015A	<R0>	0x4002_016A	<R0>	0x4002_017A	
0x4002_014B	<R0>	0x4002_015B	<R0>	0x4002_016B	<R0>	0x4002_017B	
0x4002_014C	SC3MOD0	0x4002_015C	SC3MOD2	0x4002_016C	SC3TST	0x4002_017C	
0x4002_014D	<R0>	0x4002_015D	<R0>	0x4002_016D	<R0>	0x4002_017D	
0x4002_014E	<R0>	0x4002_015E	<R0>	0x4002_016E	<R0>	0x4002_017E	
0x4002_014F	<R0>	0x4002_015F	<R0>	0x4002_016F	<R0>	0x4002_017F	

22.1.5 [5] 12-bit A/D converter (A/DC) [1/2]

<ADC B>

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4003_0200	ADBCLK	0x4003_0210	ADBCMPCR0	0x4003_0220	ADBREG0	0x4003_0230	ADBREG4
0x4003_0201	"	0x4003_0211	"	0x4003_0221	"	0x4003_0231	"
0x4003_0202	"	0x4003_0212	"	0x4003_0222	"	0x4003_0232	"
0x4003_0203	"	0x4003_0213	"	0x4003_0223	"	0x4003_0233	"
0x4003_0204	ADBMOD0	0x4003_0214	ADBCMPCR1	0x4003_0224	ADBREG1	0x4003_0234	ADBREG5
0x4003_0205	"	0x4003_0215	"	0x4003_0225	"	0x4003_0235	"
0x4003_0206	"	0x4003_0216	"	0x4003_0226	"	0x4003_0236	"
0x4003_0207	"	0x4003_0217	"	0x4003_0227	"	0x4003_0237	"
0x4003_0208	ADBMOD1	0x4003_0218	ADBCMP0	0x4003_0228	ADBREG2	0x4003_0238	ADBREG6
0x4003_0209	"	0x4003_0219	"	0x4003_0229	"	0x4003_0239	"
0x4003_020A	"	0x4003_021A	"	0x4003_022A	"	0x4003_023A	"
0x4003_020B	"	0x4003_021B	"	0x4003_022B	"	0x4003_023B	"
0x4003_020C	ADBMOD2	0x4003_021C	ADBCMP1	0x4003_022C	ADBREG3	0x4003_023C	ADBREG7
0x4003_020D	"	0x4003_021D	"	0x4003_022D	"	0x4003_023D	"
0x4003_020E	"	0x4003_021E		0x4003_022E	"	0x4003_023E	"
0x4003_020F	"	0x4003_021F		0x4003_022F	"	0x4003_023F	"

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4003_0240	ADBREG8	0x4003_0250	Reserved	0x4003_0260	Reserved	0x4003_0270	ADBPSEL8
0x4003_0241	"	0x4003_0251	Reserved	0x4003_0261	Reserved	0x4003_0271	"
0x4003_0242	"	0x4003_0252	Reserved	0x4003_0262	Reserved	0x4003_0272	"
0x4003_0243	"	0x4003_0253	Reserved	0x4003_0263	Reserved	0x4003_0273	"
0x4003_0244	ADBREG9	0x4003_0254	Reserved	0x4003_0264	Reserved	0x4003_0274	ADBPSEL9
0x4003_0245	"	0x4003_0255	Reserved	0x4003_0265	Reserved	0x4003_0275	"
0x4003_0246	"	0x4003_0256	Reserved	0x4003_0266	Reserved	0x4003_0276	"
0x4003_0247	"	0x4003_0257	Reserved	0x4003_0267	Reserved	0x4003_0277	"
0x4003_0248	ADBREG10	0x4003_0258	Reserved	0x4003_0268	ADBPSEL6	0x4003_0278	ADBPSEL10
0x4003_0249	"	0x4003_0259	Reserved	0x4003_0269	"	0x4003_0279	"
0x4003_024A	"	0x4003_025A	Reserved	0x4003_026A	"	0x4003_027A	"
0x4003_024B	"	0x4003_025B	Reserved	0x4003_026B	"	0x4003_027B	"
0x4003_024C	ADBREG11	0x4003_025C	Reserved	0x4003_026C	ADBPSEL7	0x4003_027C	ADBPSEL11
0x4003_024D	"	0x4003_025D	Reserved	0x4003_026D	"	0x4003_027D	"
0x4003_024E	"	0x4003_025E	Reserved	0x4003_026E	"	0x4003_027E	"
0x4003_024F	"	0x4003_025F	Reserved	0x4003_026F	"	0x4003_027F	"

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4003_0280	ADBPINTS0	0x4003_0290	ADBPINTS4	0x4003_02A0	ADBPSET2	0x4003_02B0	ADBTSET03
0x4003_0281	"	0x4003_0291	"	0x4003_02A1	"	0x4003_02B1	"
0x4003_0282	"	0x4003_0292	"	0x4003_02A2	"	0x4003_02B2	"
0x4003_0283	"	0x4003_0293	"	0x4003_02A3	"	0x4003_02B3	"
0x4003_0284	ADBPINTS1	0x4003_0294	ADBPINTS5	0x4003_02A4	ADBPSET3	0x4003_02B4	ADBTSET47
0x4003_0285	"	0x4003_0295	"	0x4003_02A5	"	0x4003_02B5	"
0x4003_0286	"	0x4003_0296	"	0x4003_02A6	"	0x4003_02B6	"
0x4003_0287	"	0x4003_0297	"	0x4003_02A7	"	0x4003_02B7	"
0x4003_0288	ADBPINTS2	0x4003_0298	ADBPSET0	0x4003_02A8	ADBPSET4	0x4003_02B8	ADBTSET811
0x4003_0289	"	0x4003_0299	"	0x4003_02A9	"	0x4003_02B9	"
0x4003_028A	"	0x4003_029A	"	0x4003_02AA	"	0x4003_02BA	"
0x4003_028B	"	0x4003_029B	"	0x4003_02AB	"	0x4003_02BB	"
0x4003_028C	ADBPINTS3	0x4003_029C	ADBPSET1	0x4003_02AC	ADBPSET5	0x4003_02BC	ADBSSET03
0x4003_028D	"	0x4003_029D	"	0x4003_02AD	"	0x4003_02BD	"
0x4003_028E	"	0x4003_029E	"	0x4003_02AE	"	0x4003_02BE	"
0x4003_028F	"	0x4003_029F	"	0x4003_02AF	"	0x4003_02BF	"

[5] 12-bit A/D converter (A/DC) [2/2]

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4003_02C0	ADBSSET47	0x4003_02D0	ADBASET811	0x4003_02E0		0x4003_02F0	
0x4003_02C1	"	0x4003_02D1	"	0x4003_02E1		0x4003_02F1	
0x4003_02C2	"	0x4003_02D2	"	0x4003_02E2		0x4003_02F2	
0x4003_02C3	"	0x4003_02D3	"	0x4003_02E3		0x4003_02F3	
0x4003_02C4	ADBSSET811	0x4003_02D4	Reserved	0x4003_02E4		0x4003_02F4	
0x4003_02C5	"	0x4003_02D5	Reserved	0x4003_02E5		0x4003_02F5	
0x4003_02C6	"	0x4003_02D6	Reserved	0x4003_02E6		0x4003_02F6	
0x4003_02C7	"	0x4003_02D7	Reserved	0x4003_02E7		0x4003_02F7	
0x4003_02C8	ADBASET03	0x4003_02D8		0x4003_02E8		0x4003_02F8	
0x4003_02C9	"	0x4003_02D9		0x4003_02E9		0x4003_02F9	
0x4003_02CA	"	0x4003_02DA		0x4003_02EA		0x4003_02FA	
0x4003_02CB	"	0x4003_02DB		0x4003_02EB		0x4003_02FB	
0x4003_02CC	ADBASET47	0x4003_02DC		0x4003_02EC		0x4003_02FC	
0x4003_02CD	"	0x4003_02DD		0x4003_02ED		0x4003_02FD	
0x4003_02CE	"	0x4003_02DE		0x4003_02EE		0x4003_02FE	
0x4003_02CF	"	0x4003_02DF		0x4003_02EF		0x4003_02FF	

22.1.6 [6] Watchdog timer (WDT)

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4004_0000	WDMOD	0x4004_0010		0x4004_0020		0x4004_0030	
0x4004_0001	<R0>	0x4004_0011		0x4004_0021		0x4004_0031	
0x4004_0002	<R0>	0x4004_0012		0x4004_0022		0x4004_0032	
0x4004_0003	<R0>	0x4004_0013		0x4004_0023		0x4004_0033	
0x4004_0004	WDCR	0x4004_0014		0x4004_0024		0x4004_0034	
0x4004_0005	<R0>	0x4004_0015		0x4004_0025		0x4004_0035	
0x4004_0006	<R0>	0x4004_0016		0x4004_0026		0x4004_0036	
0x4004_0007	<R0>	0x4004_0017		0x4004_0027		0x4004_0037	
0x4004_0008		0x4004_0018		0x4004_0028		0x4004_0038	
0x4004_0009		0x4004_0019		0x4004_0029		0x4004_0039	
0x4004_000A		0x4004_001A		0x4004_002A		0x4004_003A	
0x4004_000B		0x4004_001B		0x4004_002B		0x4004_003B	
0x4004_000C		0x4004_001C		0x4004_002C		0x4004_003C	
0x4004_000D		0x4004_001D		0x4004_002D		0x4004_003D	
0x4004_000E		0x4004_001E		0x4004_002E		0x4004_003E	
0x4004_000F		0x4004_001F		0x4004_002F		0x4004_003F	

22.1.7 [7] Clock generator (CG)

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4004_0200	CGSYSCR	0x4004_0210	CGCKSEL	0x4004_0220	CGIMCGA	0x4004_0230	
0x4004_0201	"	0x4004_0211	<R0>	0x4004_0221	"	0x4004_0231	
0x4004_0202	"	0x4004_0212	<R0>	0x4004_0222	"	0x4004_0232	
0x4004_0203	<R0>	0x4004_0213	<R0>	0x4004_0223	"	0x4004_0233	
0x4004_0204	CGOSCCR	0x4004_0214	CGICRCG	0x4004_0224	CGIMCGB	0x4004_0234	
0x4004_0205	"	0x4004_0215	<R0>	0x4004_0225	"	0x4004_0235	
0x4004_0206	"	0x4004_0216	<R0>	0x4004_0226	"	0x4004_0236	
0x4004_0207	"	0x4004_0217	<R0>	0x4004_0227	"	0x4004_0237	
0x4004_0208	CGSTBYCR	0x4004_0218	CGNMIFLG	0x4004_0228	CGIMCGC	0x4004_0238	
0x4004_0209	"	0x4004_0219	<R0>	0x4004_0229	"	0x4004_0239	
0x4004_020A	"	0x4004_021A	<R0>	0x4004_022A	"	0x4004_023A	
0x4004_020B	<R0>	0x4004_021B	<R0>	0x4004_022B	"	0x4004_023B	
0x4004_020C	CGPLLSEL	0x4004_021C	CGRSTFLG	0x4004_022C	CGIMCGD	0x4004_023C	
0x4004_020D	"	0x4004_021D	<R0>	0x4004_022D	"	0x4004_023D	
0x4004_020E	<R0>	0x4004_021E	<R0>	0x4004_022E	"	0x4004_023E	
0x4004_020F	<R0>	0x4004_021F	<R0>	0x4004_022F	"	0x4004_023F	

22.1.8   [8] Oscillation frequency detector (OFD)

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4004_0800	OFDCR1	0x4004_0810	CLKSMX PLLOFF	0x4004_0820		0x4004_0830	
0x4004_0801	<R0>	0x4004_0811	"	0x4004_0821		0x4004_0831	
0x4004_0802	<R0>	0x4004_0812	<R0>	0x4004_0822		0x4004_0832	
0x4004_0803	<R0>	0x4004_0813	<R0>	0x4004_0823		0x4004_0833	
0x4004_0804	OFDCR2	0x4004_0814	CLKSMX PLLON	0x4004_0824		0x4004_0834	
0x4004_0805	<R0>	0x4004_0815	"	0x4004_0825		0x4004_0835	
0x4004_0806	<R0>	0x4004_0816	<R0>	0x4004_0826		0x4004_0836	
0x4004_0807	<R0>	0x4004_0817	<R0>	0x4004_0827		0x4004_0837	
0x4004_0808	OFDMIN PLLOFF	0x4004_0818		0x4004_0828		0x4004_0838	
0x4004_0809	"	0x4004_0819		0x4004_0829		0x4004_0839	
0x4004_080A	<R0>	0x4004_081A		0x4004_082A		0x4004_083A	
0x4004_080B	<R0>	0x4004_081B		0x4004_082B		0x4004_083B	
0x4004_080C	CLKSMIN PLLON	0x4004_081C		0x4004_082C		0x4004_083C	
0x4004_080D	"	0x4004_081D		0x4004_082D		0x4004_083D	
0x4004_080E	<R0>	0x4004_081E		0x4004_082E		0x4004_083E	
0x4004_080F	<R0>	0x4004_081F		0x4004_082F		0x4004_083F	

22.1.9   [9] Power on reset (POR), Voltage detecting circuit (VLTD)

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4004_0900	VDCR	0x4004_0910		0x4004_0920		0x4004_0930	
0x4004_0901		0x4004_0911		0x4004_0921		0x4004_0931	
0x4004_0902		0x4004_0912		0x4004_0922		0x4004_0932	
0x4004_0903		0x4004_0913		0x4004_0923		0x4004_0933	
0x4004_0904	Reserved	0x4004_0914		0x4004_0924		0x4004_0934	
0x4004_0905		0x4004_0915		0x4004_0925		0x4004_0935	
0x4004_0906		0x4004_0916		0x4004_0926		0x4004_0936	
0x4004_0907		0x4004_0917		0x4004_0927		0x4004_0937	
0x4004_0908		0x4004_0918		0x4004_0928		0x4004_0938	
0x4004_0909		0x4004_0919		0x4004_0929		0x4004_0939	
0x4004_090A		0x4004_091A		0x4004_092A		0x4004_093A	
0x4004_090B		0x4004_091B		0x4004_092B		0x4004_093B	
0x4004_090C		0x4004_091C		0x4004_092C		0x4004_093C	
0x4004_090D		0x4004_091D		0x4004_092D		0x4004_093D	
0x4004_090E		0x4004_091E		0x4004_092E		0x4004_093E	
0x4004_090F		0x4004_091F		0x4004_092F		0x4004_093F	

22.1.10 [10] Vector engine (VE) [1/3]

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4005_0000	VEEN	0x4005_0010	VEREPTIME	0x4005_0020	VEERRDET	0x4005_0030	VETMPREG1
0x4005_0001	<R0>	0x4005_0011	<R0>	0x4005_0021	<R0>	0x4005_0031	"
0x4005_0002	<R0>	0x4005_0012	<R0>	0x4005_0022	<R0>	0x4005_0032	"
0x4005_0003	<R0>	0x4005_0013	<R0>	0x4005_0023	<R0>	0x4005_0033	"
0x4005_0004	VECPURUNTRG	0x4005_0014	VETRGMODE	0x4005_0024	VESCHTASKRUN	0x4005_0034	VETMPREG2
0x4005_0005	<R0>	0x4005_0015	<R0>	0x4005_0025	"	0x4005_0035	"
0x4005_0006	<R0>	0x4005_0016	<R0>	0x4005_0026	<R0>	0x4005_0036	"
0x4005_0007	<R0>	0x4005_0017	<R0>	0x4005_0027	<R0>	0x4005_0037	"
0x4005_0008	VETASKAPP	0x4005_0018	VEERRINTEN	0x4005_0028	Reserved	0x4005_0038	VETMPREG3
0x4005_0009	<R0>	0x4005_0019	<R0>	0x4005_0029	Reserved	0x4005_0039	"
0x4005_000A	<R0>	0x4005_001A	<R0>	0x4005_002A	Reserved	0x4005_003A	"
0x4005_000B	<R0>	0x4005_001B	<R0>	0x4005_002B	Reserved	0x4005_003B	"
0x4005_000C	VEACTSCH	0x4005_001C	VECOMPEND	0x4005_002C	VETMPREG0	0x4005_003C	VETMPREG4
0x4005_000D	<R0>	0x4005_001D	<R0>	0x4005_002D	"	0x4005_003D	"
0x4005_000E	<R0>	0x4005_001E	<R0>	0x4005_002E	"	0x4005_003E	"
0x4005_000F	<R0>	0x4005_001F	<R0>	0x4005_002F	"	0x4005_003F	"

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4005_0040	VETMPREG5	0x4005_0050	Reserved	0x4005_0060	Reserved	0x4005_0070	Reserved
0x4005_0041	"	0x4005_0051	Reserved	0x4005_0061	Reserved	0x4005_0071	Reserved
0x4005_0042	"	0x4005_0052	Reserved	0x4005_0062	Reserved	0x4005_0072	Reserved
0x4005_0043	"	0x4005_0053	Reserved	0x4005_0063	Reserved	0x4005_0073	Reserved
0x4005_0044	Reserved	0x4005_0054	Reserved	0x4005_0064	Reserved	0x4005_0074	Reserved
0x4005_0045	Reserved	0x4005_0055	Reserved	0x4005_0065	Reserved	0x4005_0075	Reserved
0x4005_0046	Reserved	0x4005_0056	Reserved	0x4005_0066	Reserved	0x4005_0076	Reserved
0x4005_0047	Reserved	0x4005_0057	Reserved	0x4005_0067	Reserved	0x4005_0077	Reserved
0x4005_0048	Reserved	0x4005_0058	Reserved	0x4005_0068	Reserved	0x4005_0078	Reserved
0x4005_0049	Reserved	0x4005_0059	Reserved	0x4005_0069	Reserved	0x4005_0079	Reserved
0x4005_004A	Reserved	0x4005_005A	Reserved	0x4005_006A	Reserved	0x4005_007A	Reserved
0x4005_004B	Reserved	0x4005_005B	Reserved	0x4005_006B	Reserved	0x4005_007B	Reserved
0x4005_004C	Reserved	0x4005_005C	Reserved	0x4005_006C	Reserved	0x4005_007C	Reserved
0x4005_004D	Reserved	0x4005_005D	Reserved	0x4005_006D	Reserved	0x4005_007D	Reserved
0x4005_004E	Reserved	0x4005_005E	Reserved	0x4005_006E	Reserved	0x4005_007E	Reserved
0x4005_004F	Reserved	0x4005_005F	Reserved	0x4005_006F	Reserved	0x4005_007F	Reserved

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x4005_0080	Reserved	0x4005_0090	Reserved	0x4005_00A0	Reserved	0x4005_00B0	Reserved
0x4005_0081	Reserved	0x4005_0091	Reserved	0x4005_00A1	Reserved	0x4005_00B1	Reserved
0x4005_0082	Reserved	0x4005_0092	Reserved	0x4005_00A2	Reserved	0x4005_00B2	Reserved
0x4005_0083	Reserved	0x4005_0093	Reserved	0x4005_00A3	Reserved	0x4005_00B3	Reserved
0x4005_0084	Reserved	0x4005_0094	Reserved	0x4005_00A4	Reserved	0x4005_00B4	Reserved
0x4005_0085	Reserved	0x4005_0095	Reserved	0x4005_00A5	Reserved	0x4005_00B5	Reserved
0x4005_0086	Reserved	0x4005_0096	Reserved	0x4005_00A6	Reserved	0x4005_00B6	Reserved
0x4005_0087	Reserved	0x4005_0097	Reserved	0x4005_00A7	Reserved	0x4005_00B7	Reserved
0x4005_0088	Reserved	0x4005_0098	Reserved	0x4005_00A8	Reserved	0x4005_00B8	Reserved
0x4005_0089	Reserved	0x4005_0099	Reserved	0x4005_00A9	Reserved	0x4005_00B9	Reserved
0x4005_008A	Reserved	0x4005_009A	Reserved	0x4005_00AA	Reserved	0x4005_00BA	Reserved
0x4005_008B	Reserved	0x4005_009B	Reserved	0x4005_00AB	Reserved	0x4005_00BB	Reserved
0x4005_008C	Reserved	0x4005_009C	Reserved	0x4005_00AC	Reserved	0x4005_00BC	Reserved
0x4005_008D	Reserved	0x4005_009D	Reserved	0x4005_00AD	Reserved	0x4005_00BD	Reserved
0x4005_008E	Reserved	0x4005_009E	Reserved	0x4005_00AE	Reserved	0x4005_00BE	Reserved
0x4005_008F	Reserved	0x4005_009F	Reserved	0x4005_00AF	Reserved	0x4005_00BF	Reserved

Note: Do not access to "Reserved".

[10] Vector engine (VE) [2/3]

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4005_00C0	Reserved		0x4005_00D0	Reserved		0x4005_00E0	VEMODE1		0x4005_00F0	VETHETA1
0x4005_00C1	Reserved		0x4005_00D1	Reserved		0x4005_00E1	<R0>		0x4005_00F1	"
0x4005_00C2	Reserved		0x4005_00D2	Reserved		0x4005_00E2	<R0>		0x4005_00F2	<R0>
0x4005_00C3	Reserved		0x4005_00D3	Reserved		0x4005_00E3	<R0>		0x4005_00F3	<R0>
0x4005_00C4	Reserved		0x4005_00D4	Reserved		0x4005_00E4	VEFMODE1		0x4005_00F4	VEIDREF1
0x4005_00C5	Reserved		0x4005_00D5	Reserved		0x4005_00E5	"		0x4005_00F5	"
0x4005_00C6	Reserved		0x4005_00D6	Reserved		0x4005_00E6	<R0>		0x4005_00F6	<R0>
0x4005_00C7	Reserved		0x4005_00D7	Reserved		0x4005_00E7	<R0>		0x4005_00F7	<R0>
0x4005_00C8	Reserved		0x4005_00D8	Reserved		0x4005_00E8	VETPWM1		0x4005_00F8	VEIQREF1
0x4005_00C9	Reserved		0x4005_00D9	Reserved		0x4005_00E9	"		0x4005_00F9	"
0x4005_00CA	Reserved		0x4005_00DA	Reserved		0x4005_00EA	<R0>		0x4005_00FA	<R0>
0x4005_00CB	Reserved		0x4005_00DB	Reserved		0x4005_00EB	<R0>		0x4005_00FB	<R0>
0x4005_00CC	Reserved		0x4005_00DC	VEMCTLF1		0x4005_00EC	VEOMEGA1		0x4005_00FC	VEVD1
0x4005_00CD	Reserved		0x4005_00DD	<R0>		0x4005_00ED	"		0x4005_00FD	<R0>
0x4005_00CE	Reserved		0x4005_00DE	<R0>		0x4005_00EE	<R0>		0x4005_00FE	<R0>
0x4005_00CF	Reserved		0x4005_00DF	<R0>		0x4005_00EF	<R0>		0x4005_00FF	<R0>

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4005_0100	VEVQ1		0x4005_0110	VECIQKP1		0x4005_0120	VEVQILH1		0x4005_0130	VETRGCRC1
0x4005_0101	<R0>		0x4005_0111	"		0x4005_0121	"		0x4005_0131	"
0x4005_0102	<R0>		0x4005_0112	<R0>		0x4005_0122	<R0>		0x4005_0132	<R0>
0x4005_0103	<R0>		0x4005_0113	<R0>		0x4005_0123	<R0>		0x4005_0133	<R0>
0x4005_0104	VECIDKI1		0x4005_0114	VEVDIH1		0x4005_0124	VEFPWMCHG1		0x4005_0134	Reserved
0x4005_0105	"		0x4005_0115	"		0x4005_0125	"		0x4005_0135	Reserved
0x4005_0106	<R0>		0x4005_0116	<R0>		0x4005_0126	<R0>		0x4005_0136	Reserved
0x4005_0107	<R0>		0x4005_0117	<R0>		0x4005_0127	<R0>		0x4005_0137	Reserved
0x4005_0108	VECIDKP1		0x4005_0118	VEVDILH1		0x4005_0128	VEVMDPRD1		0x4005_0138	VECOS1
0x4005_0109	"		0x4005_0119	"		0x4005_0129	"		0x4005_0139	"
0x4005_010A	<R0>		0x4005_011A	<R0>		0x4005_012A	<R0>		0x4005_013A	<R0>
0x4005_010B	<R0>		0x4005_011B	<R0>		0x4005_012B	<R0>		0x4005_013B	<R0>
0x4005_010C	VECIQKI1		0x4005_011C	VEVQIH1		0x4005_012C	VEMINPLS0		0x4005_013C	VESIN1
0x4005_010D	"		0x4005_011D	"		0x4005_012D	"		0x4005_013D	"
0x4005_010E	<R0>		0x4005_011E	<R0>		0x4005_012E	<R0>		0x4005_013E	<R0>
0x4005_010F	<R0>		0x4005_011F	<R0>		0x4005_012F	<R0>		0x4005_013F	<R0>

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4005_0140	VECOSM1		0x4005_0150	VEIAO1		0x4005_0160	VEIBADC1		0x4005_0170	VEIQ1
0x4005_0141	"		0x4005_0151	"		0x4005_0161	"		0x4005_0171	"
0x4005_0142	<R0>		0x4005_0152	<R0>		0x4005_0162	<R0>		0x4005_0172	"
0x4005_0143	<R0>		0x4005_0153	<R0>		0x4005_0163	<R0>		0x4005_0173	"
0x4005_0144	VESINM1		0x4005_0154	VEIBO1		0x4005_0164	VEICADC1		0x4005_0174	Reserved
0x4005_0145	"		0x4005_0155	"		0x4005_0165	"		0x4005_0175	Reserved
0x4005_0146	<R0>		0x4005_0156	<R0>		0x4005_0166	<R0>		0x4005_0176	Reserved
0x4005_0147	<R0>		0x4005_0157	<R0>		0x4005_0167	<R0>		0x4005_0177	Reserved
0x4005_0148	VESECTOR1		0x4005_0158	VEICO1		0x4005_0168	VEVDC1		0x4005_0178	VEADC
0x4005_0149	<R0>		0x4005_0159	"		0x4005_0169	"		0x4005_0179	"
0x4005_014A	<R0>		0x4005_015A	<R0>		0x4005_016A	<R0>		0x4005_017A	<R0>
0x4005_014B	<R0>		0x4005_015B	<R0>		0x4005_016B	<R0>		0x4005_017B	<R0>
0x4005_014C	VESECTORM1		0x4005_015C	VEIAADC1		0x4005_016C	VEID1		0x4005_017C	Reserved
0x4005_014D	<R0>		0x4005_015D	"		0x4005_016D	"		0x4005_017D	Reserved
0x4005_014E	<R0>		0x4005_015E	<R0>		0x4005_016E	"		0x4005_017E	Reserved
0x4005_014F	<R0>		0x4005_015F	<R0>		0x4005_016F	"		0x4005_017F	Reserved

Note: Do not access to "Reserved".

[10] Vector engine (VE) [3/3]

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4005_0180	Reserved		0x4005_0190	Reserved		0x4005_01A0	VEVCMPV1		0x4005_01B0	VEVTRGCMP11
0x4005_0181	Reserved		0x4005_0191	Reserved		0x4005_01A1	"		0x4005_01B1	"
0x4005_0182	Reserved		0x4005_0192	Reserved		0x4005_01A2	<R0>		0x4005_01B2	<R0>
0x4005_0183	Reserved		0x4005_0193	Reserved		0x4005_01A3	<R0>		0x4005_01B3	<R0>
0x4005_0184	Reserved		0x4005_0194	Reserved		0x4005_01A4	VEVCMPW1		0x4005_01B4	VEVTRGSEL1
0x4005_0185	Reserved		0x4005_0195	Reserved		0x4005_01A5	"		0x4005_01B5	<R0>
0x4005_0186	Reserved		0x4005_0196	Reserved		0x4005_01A6	<R0>		0x4005_01B6	<R0>
0x4005_0187	Reserved		0x4005_0197	Reserved		0x4005_01A7	<R0>		0x4005_01B7	<R0>
0x4005_0188	Reserved		0x4005_0198	Reserved		0x4005_01A8	VEOUTCR1		0x4005_01B8	VEEMGRS1
0x4005_0189	Reserved		0x4005_0199	Reserved		0x4005_01A9	"		0x4005_01B9	<R0>
0x4005_018A	Reserved		0x4005_019A	Reserved		0x4005_01AA	<R0>		0x4005_01BA	<R0>
0x4005_018B	Reserved		0x4005_019B	Reserved		0x4005_01AB	<R0>		0x4005_01BB	<R0>
0x4005_018C	Reserved		0x4005_019C	VEVCMPU1		0x4005_01AC	VEVTRGCMP01		0x4005_01BC	Reserved
0x4005_018D	Reserved		0x4005_019D	"		0x4005_01AD	"		0x4005_01BD	Reserved
0x4005_018E	Reserved		0x4005_019E	<R0>		0x4005_01AE	<R0>		0x4005_01BE	Reserved
0x4005_018F	Reserved		0x4005_019F	<R0>		0x4005_01AF	<R0>		0x4005_01BF	Reserved

ADR	Register name		ADR	Register name		ADR	Register name		ADR	Register name
0x4005_01C0	Reserved		0x4005_01D0	Reserved		0x4005_01E0	Reserved		0x4005_01F0	Reserved
0x4005_01C1	Reserved		0x4005_01D1	Reserved		0x4005_01E1	Reserved		0x4005_01F1	Reserved
0x4005_01C2	Reserved		0x4005_01D2	Reserved		0x4005_01E2	Reserved		0x4005_01F2	Reserved
0x4005_01C3	Reserved		0x4005_01D3	Reserved		0x4005_01E3	Reserved		0x4005_01F3	Reserved
0x4005_01C4	Reserved		0x4005_01D4	Reserved		0x4005_01E4	Reserved		0x4005_01F4	Reserved
0x4005_01C5	Reserved		0x4005_01D5	Reserved		0x4005_01E5	Reserved		0x4005_01F5	Reserved
0x4005_01C6	Reserved		0x4005_01D6	Reserved		0x4005_01E6	Reserved		0x4005_01F6	Reserved
0x4005_01C7	Reserved		0x4005_01D7	Reserved		0x4005_01E7	Reserved		0x4005_01F7	Reserved
0x4005_01C8	Reserved		0x4005_01D8	Reserved		0x4005_01E8	Reserved		0x4005_01F8	Reserved
0x4005_01C9	Reserved		0x4005_01D9	Reserved		0x4005_01E9	Reserved		0x4005_01F9	Reserved
0x4005_01CA	Reserved		0x4005_01DA	Reserved		0x4005_01EA	Reserved		0x4005_01FA	Reserved
0x4005_01CB	Reserved		0x4005_01DB	Reserved		0x4005_01EB	Reserved		0x4005_01FB	Reserved
0x4005_01CC	Reserved		0x4005_01DC	Reserved		0x4005_01EC	Reserved		0x4005_01FC	Reserved
0x4005_01CD	Reserved		0x4005_01DD	Reserved		0x4005_01ED	Reserved		0x4005_01FD	Reserved
0x4005_01CE	Reserved		0x4005_01DE	Reserved		0x4005_01EE	Reserved		0x4005_01FE	Reserved
0x4005_01CF	Reserved		0x4005_01DF	Reserved		0x4005_01EF	Reserved		0x4005_01FF	

Note: Do not access to "Reserved".

22.1.11 [11] Programmable motor driver (PMD)

ADR	Register name
0x4005_0480	PMD1MDEN
0x4005_0481	<R0>
0x4005_0482	<R0>
0x4005_0483	<R0>
0x4005_0484	PMD1PORTMD
0x4005_0485	<R0>
0x4005_0486	<R0>
0x4005_0487	<R0>
0x4005_0488	PMD1MDCR
0x4005_0489	<R0>
0x4005_048A	<R0>
0x4005_048B	<R0>
0x4005_048C	PMD1CNTSTA
0x4005_048D	<R0>
0x4005_048E	<R0>
0x4005_048F	<R0>

ADR	Register name
0x4005_0490	PMD1MDCNT
0x4005_0491	"
0x4005_0492	<R0>
0x4005_0493	<R0>
0x4005_0494	PMD1MDPRD
0x4005_0495	"
0x4005_0496	<R0>
0x4005_0497	<R0>
0x4005_0498	PMD1CMPU
0x4005_0499	"
0x4005_049A	<R0>
0x4005_049B	<R0>
0x4005_049C	PMD1CMPV
0x4005_049D	"
0x4005_049E	<R0>
0x4005_049F	<R0>

ADR	Register name
0x4005_04A0	PMD1CMPW
0x4005_04A1	"
0x4005_04A2	<R0>
0x4005_04A3	<R0>
0x4005_04A4	PMD1MODESEL
0x4005_04A5	<R0>
0x4005_04A6	<R0>
0x4005_04A7	<R0>
0x4005_04A8	PMD1MDOUT
0x4005_04A9	"
0x4005_04AA	<R0>
0x4005_04AB	<R0>
0x4005_04AC	PMD1MDPOT
0x4005_04AD	<R0>
0x4005_04AE	<R0>
0x4005_04AF	<R0>

ADR	Register name
0x4005_04B0	PMD1EMGREL
0x4005_04B1	<R0>
0x4005_04B2	<R0>
0x4005_04B3	<R0>
0x4005_04B4	PMD1EMGCR
0x4005_04B5	"
0x4005_04B6	<R0>
0x4005_04B7	<R0>
0x4005_04B8	PMD1EMGSTA
0x4005_04B9	<R0>
0x4005_04BA	<R0>
0x4005_04BB	<R0>
0x4005_04BC	PMD1OVVCR
0x4005_04BD	"
0x4005_04BE	<R0>
0x4005_04BF	<R0>

ADR	Register name
0x4005_04C0	PMD1OVVSTA
0x4005_04C1	<R0>
0x4005_04C2	<R0>
0x4005_04C3	<R0>
0x4005_04C4	PMD1DTR
0x4005_04C5	<R0>
0x4005_04C6	<R0>
0x4005_04C7	<R0>
0x4005_04C8	PMD1TRGCMP0
0x4005_04C9	"
0x4005_04CA	<R0>
0x4005_04CB	<R0>
0x4005_04CC	PMD1TRGCMP1
0x4005_04CD	"
0x4005_04CE	<R0>
0x4005_04CF	<R0>

ADR	Register name
0x4005_04D0	PMD1TRGCMP2
0x4005_04D1	"
0x4005_04D2	<R0>
0x4005_04D3	<R0>
0x4005_04D4	PMD1TRGCMP3
0x4005_04D5	"
0x4005_04D6	<R0>
0x4005_04D7	<R0>
0x4005_04D8	PMD1TRGCR
0x4005_04D9	"
0x4005_04DA	<R0>
0x4005_04DB	<R0>
0x4005_04DC	PMD1TRGMD
0x4005_04DD	"
0x4005_04DE	"
0x4005_04DF	"

ADR	Register name
0x4005_04E0	PMD1TRGSEL
0x4005_04E1	"
0x4005_04E2	"
0x4005_04E3	"
0x4005_04E4	Reserved
0x4005_04E5	Reserved
0x4005_04E6	Reserved
0x4005_04E7	Reserved
0x4005_04E8	Reserved
0x4005_04E9	Reserved
0x4005_04EA	Reserved
0x4005_04EB	Reserved
0x4005_04EC	Reserved
0x4005_04ED	Reserved
0x4005_04EE	Reserved
0x4005_04EF	Reserved

ADR	Register name
0x4005_04F0	Reserved
0x4005_04F1	Reserved
0x4005_04F2	Reserved
0x4005_04F3	Reserved
0x4005_04F4	Reserved
0x4005_04F5	Reserved
0x4005_04F6	Reserved
0x4005_04F7	Reserved
0x4005_04F8	Reserved
0x4005_04F9	Reserved
0x4005_04FA	Reserved
0x4005_04FB	Reserved
0x4005_04FC	Reserved
0x4005_04FD	Reserved
0x4005_04FE	Reserved
0x4005_04FF	Reserved

TOSHIBA

22.1.12 [12] Flash

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x41FF_F000	Reserved	0x41FF_F010	FCSECBIT	0x41FF_F020	FCFLCS	0x41FF_F030	Reserved
0x41FF_F001	Reserved	0x41FF_F011	<R0>	0x41FF_F021	"	0x41FF_F031	Reserved
0x41FF_F002	Reserved	0x41FF_F012	<R0>	0x41FF_F022	"	0x41FF_F032	Reserved
0x41FF_F003	Reserved	0x41FF_F013	<R0>	0x41FF_F023	"	0x41FF_F033	Reserved
0x41FF_F004	Reserved	0x41FF_F014	Reserved	0x41FF_F024	Reserved	0x41FF_F034	Reserved
0x41FF_F005	Reserved	0x41FF_F015	Reserved	0x41FF_F025	Reserved	0x41FF_F035	Reserved
0x41FF_F006	Reserved	0x41FF_F016	Reserved	0x41FF_F026	Reserved	0x41FF_F036	Reserved
0x41FF_F007	Reserved	0x41FF_F017	Reserved	0x41FF_F027	Reserved	0x41FF_F037	Reserved
0x41FF_F008	Reserved	0x41FF_F018	Reserved	0x41FF_F028	Reserved	0x41FF_F038	Reserved
0x41FF_F009	Reserved	0x41FF_F019	Reserved	0x41FF_F029	Reserved	0x41FF_F039	Reserved
0x41FF_F00A	Reserved	0x41FF_F01A	Reserved	0x41FF_F02A	Reserved	0x41FF_F03A	Reserved
0x41FF_F00B	Reserved	0x41FF_F01B	Reserved	0x41FF_F02B	Reserved	0x41FF_F03B	Reserved
0x41FF_F00C	Reserved	0x41FF_F01C	Reserved	0x41FF_F02C	Reserved	0x41FF_F03C	Reserved
0x41FF_F00D	Reserved	0x41FF_F01D	Reserved	0x41FF_F02D	Reserved	0x41FF_F03D	Reserved
0x41FF_F00E	Reserved	0x41FF_F01E	Reserved	0x41FF_F02E	Reserved	0x41FF_F03E	Reserved
0x41FF_F00F	Reserved	0x41FF_F01F	Reserved	0x41FF_F02F	Reserved	0x41FF_F03F	Reserved

22.1.13 [13] Reserved area

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x41FF_F040	Reserved	0x41FF_F050	Reserved	0x41FF_F060	Reserved	0x41FF_F070	Reserved
0x41FF_F041	Reserved	0x41FF_F051	Reserved	0x41FF_F061	Reserved	0x41FF_F071	Reserved
0x41FF_F042	Reserved	0x41FF_F052	Reserved	0x41FF_F062	Reserved	0x41FF_F072	Reserved
0x41FF_F043	Reserved	0x41FF_F053	Reserved	0x41FF_F063	Reserved	0x41FF_F073	Reserved
0x41FF_F044	Reserved	0x41FF_F054	Reserved	0x41FF_F064	Reserved	0x41FF_F074	Reserved
0x41FF_F045	Reserved	0x41FF_F055	Reserved	0x41FF_F065	Reserved	0x41FF_F075	Reserved
0x41FF_F046	Reserved	0x41FF_F056	Reserved	0x41FF_F066	Reserved	0x41FF_F076	Reserved
0x41FF_F047	Reserved	0x41FF_F057	Reserved	0x41FF_F067	Reserved	0x41FF_F077	Reserved
0x41FF_F048	Reserved	0x41FF_F058	Reserved	0x41FF_F068	Reserved	0x41FF_F078	Reserved
0x41FF_F049	Reserved	0x41FF_F059	Reserved	0x41FF_F069	Reserved	0x41FF_F079	Reserved
0x41FF_F04A	Reserved	0x41FF_F05A	Reserved	0x41FF_F06A	Reserved	0x41FF_F07A	Reserved
0x41FF_F04B	Reserved	0x41FF_F05B	Reserved	0x41FF_F06B	Reserved	0x41FF_F07B	Reserved
0x41FF_F04C	Reserved	0x41FF_F05C	Reserved	0x41FF_F06C	Reserved	0x41FF_F07C	Reserved
0x41FF_F04D	Reserved	0x41FF_F05D	Reserved	0x41FF_F06D	Reserved	0x41FF_F07D	Reserved
0x41FF_F04E	Reserved	0x41FF_F05E	Reserved	0x41FF_F06E	Reserved	0x41FF_F07E	Reserved
0x41FF_F04F	Reserved	0x41FF_F05F	Reserved	0x41FF_F06F	Reserved	0x41FF_F07F	Reserved

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
0x41FF_F080	Reserved	0x41FF_F090	Reserved	0x41FF_F0A0	Reserved	0x41FF_F0B0	Reserved
0x41FF_F081	Reserved	0x41FF_F091	Reserved	0x41FF_F0A1	Reserved	0x41FF_F0B1	Reserved
0x41FF_F082	Reserved	0x41FF_F092	Reserved	0x41FF_F0A2	Reserved	0x41FF_F0B2	Reserved
0x41FF_F083	Reserved	0x41FF_F093	Reserved	0x41FF_F0A3	Reserved	0x41FF_F0B3	Reserved
0x41FF_F084	Reserved	0x41FF_F094	Reserved	0x41FF_F0A4	Reserved	0x41FF_F0B4	Reserved
0x41FF_F085	Reserved	0x41FF_F095	Reserved	0x41FF_F0A5	Reserved	0x41FF_F0B5	Reserved
0x41FF_F086	Reserved	0x41FF_F096	Reserved	0x41FF_F0A6	Reserved	0x41FF_F0B6	Reserved
0x41FF_F087	Reserved	0x41FF_F097	Reserved	0x41FF_F0A7	Reserved	0x41FF_F0B7	Reserved
0x41FF_F088	Reserved	0x41FF_F098	Reserved	0x41FF_F0A8	Reserved	0x41FF_F0B8	Reserved
0x41FF_F089	Reserved	0x41FF_F099	Reserved	0x41FF_F0A9	Reserved	0x41FF_F0B9	Reserved
0x41FF_F08A	Reserved	0x41FF_F09A	Reserved	0x41FF_F0AA	Reserved	0x41FF_F0BA	Reserved
0x41FF_F08B	Reserved	0x41FF_F09B	Reserved	0x41FF_F0AB	Reserved	0x41FF_F0BB	Reserved
0x41FF_F08C	Reserved	0x41FF_F09C	Reserved	0x41FF_F0AC	Reserved	0x41FF_F0BC	Reserved
0x41FF_F08D	Reserved	0x41FF_F09D	Reserved	0x41FF_F0AD	Reserved	0x41FF_F0BD	Reserved
0x41FF_F08E	Reserved	0x41FF_F09E	Reserved	0x41FF_F0AE	Reserved	0x41FF_F0BE	Reserved
0x41FF_F08F	Reserved	0x41FF_F09F	Reserved	0x41FF_F0AF	Reserved	0x41FF_F0BF	Reserved

23. Port Section Equivalent Circuit Schematics

- How to read the schematics

   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 ohms to several hundreds of ohms. Damping resistor and Feedback resistor are shown with a typical value.

- PA0-7, PB0-7, PD4-6, PE0-7, PF0-4, PG0-7, PM0-1
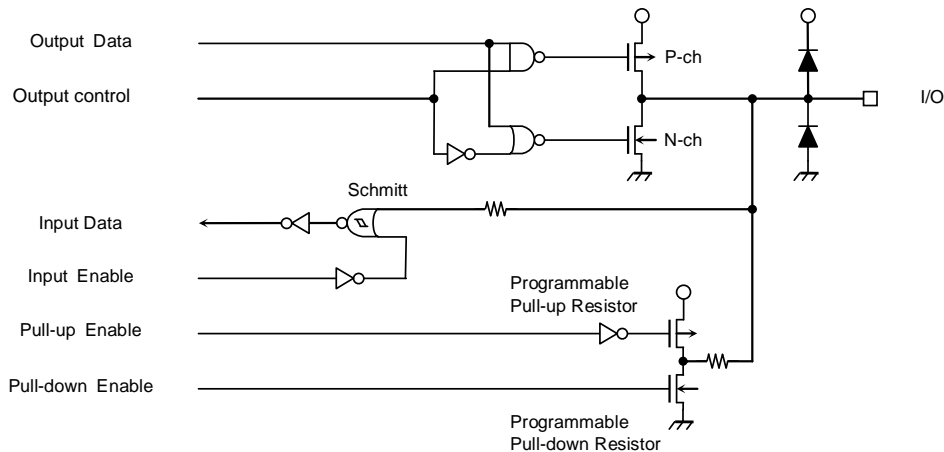
Note1: TMPM373 doesn't have PA0, PA1, PA3, PB7, PB2, PB1, PB0, PD4, PD5, PE3 and PE5 pins.
Note2: TMPM374 doesn't have PA0, PA1, PA2, PA3, PB7, PB2, PB1, PB0, PD4-6, PE3, PE5 and PF1 pins.

- PJ0-7, PK0-1, PI3
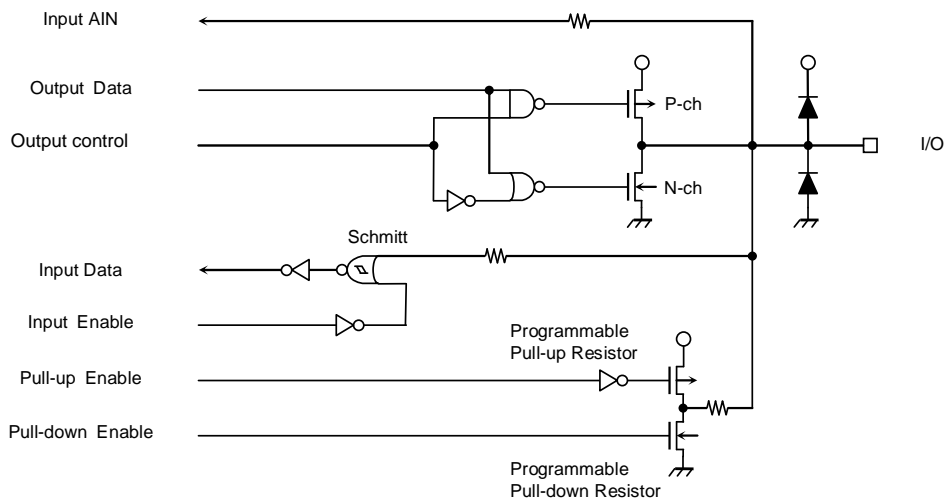
Note1: TMPM373 doesn't have PJ1, PJ2, PJ3 and PJ4 pins.
Note2: TMPM374 doesn't have PJ1, PJ2, PJ3, PJ4 and PJ5 pins.

- X1, X2
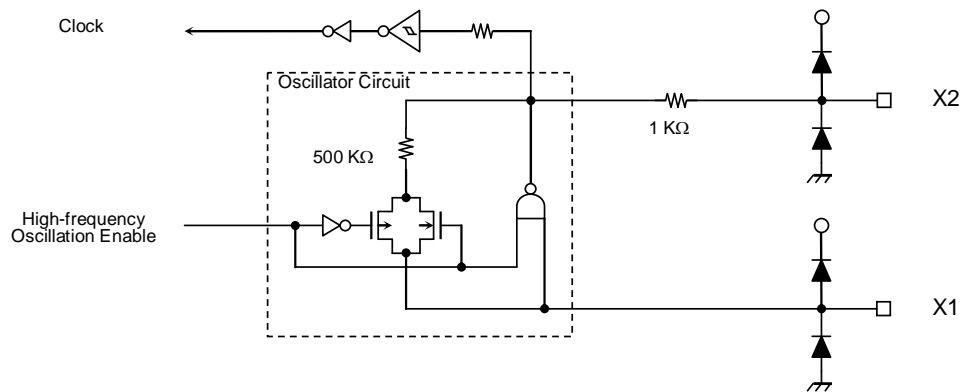
- $\overline{RESET}$
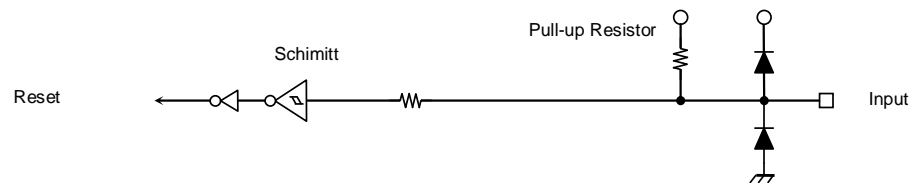
- MODE
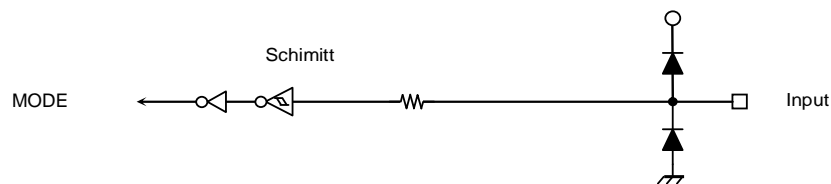
Note : MODE must be connected with GND.

- AVDD5B / VREFHB , AVSSB / VREFLB

Equivalent Circuit
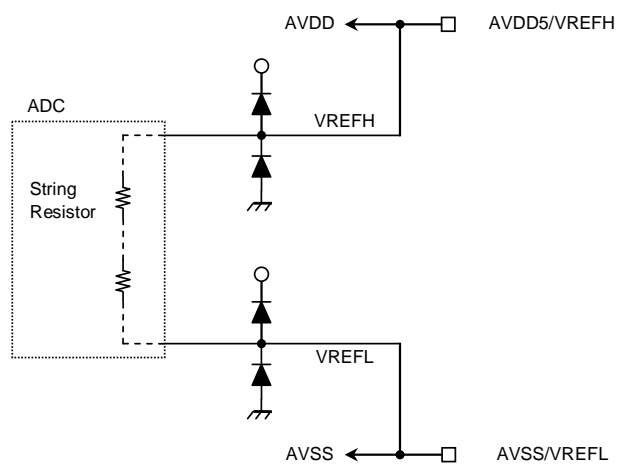
24 Electrical Characteristics

24.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5	-0.3 to 6	V
		RVDD5	-0.3 to 6	
		AVDD5B	-0.3 to 6	
Capacitor voltage		VOUT15	-0.3 to 3	
		VOUT3	-0.3 to 3.9	
Input voltage		VIN	-0.3 to VDD+0.3(Note1)	
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	-5	
	Total	ΣI_{OH}	-50	
Power consumption		PD	600 (Ta = 85°C) 350 (Ta = 105°C)	mW
Soldering temperature (10s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	-55 to 125	°C
Operating Temperature		T_{OPR}	-40 to 105	°C

(Note) **Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.**

(Note 1) VDD = DVDD5 = AVDD5B = RVDD5

24.2 DC Electrical Characteristics (1/2)

DVSS = AVSSB = 0V , Ta= −40 to 105°C

Parameter		Symbol	Rating		Min	Typ. (Note 1)	Max	Unit
Supply voltage (Note2)		DVDD5 AVDD5B RVDD5	fosc = 8 to 10MHz fsys = 1 to 80MHz		4.5	−	5.5	V
Supply voltage (during Flash W/E) (Note2)		DVDD5 AVDD5B RVDD5	fosc = 8 to 10MHz fsys = 1 to 80MHz （Ta（°C）= 0 to 70）		4.5	−	5.5	
Low-level input voltage	Schmitt-Input	V_{IL1}	VDD = 4.5 to 5.5V		−0.3	−	0.25 VDD	
Hight-level input voltage	Schmitt-Input	V_{IH1}	VDD = 4.5V to 5.5V		0.75 VDD		VDD	
Capacitance for VOUT15 and VOUT3 (Note 3)		Cout	RVDD5 = 4.5V to 5.5V VOUT15,VOUT3		3.3	−	4.7	μF
Low-level output voltage		V_{OL}	I_{OL}= 1.6mA	VDD≧4.5V (Note4)	−	−	0.4	V
High-level output voltage		V_{OH}	I_{OH}=−1.6mA	VDD≧4.5V (Note4)	4.1	−	−	V
Input leakage current		I_{LI}	0.0 ≦ V_{IN} ≦ VDD (Note4)		−	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≦ V_{IN} ≦ VDD − 0.2 (Note4)		−	0.05	±10	
Pull-up resister at Reset		R_{RST}	4.5V ≦ VDD ≦ 5.5V (Note4)		−	50	150	kΩ
Programmable pull-up/ pull-down resistor		P_{KH}	4.5V ≦ VDD ≦ 5.5V (Note4)		−	50	150	kΩ
Schmitt-Triggered port		V_{TH}	4.5V ≦ VDD ≦ 5.5V (Note4)		0.3	0.6	−	V
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1MHz		−	−	10	pF

(Note 1) Ta = 25°C, DVDD5 = AVDD5B = RVDD5 = 5V, unless otherwise noted.

(Note 2) The same voltage must be supplied to DVDD5, AVDD5B and RVDD5.

(Note 3) VOUT15 and VOUT3 pin should be connected to GND via same value of capacitance. The IC outside can not have the power supply from VOUT15 and VOUT3.

(Note4) VDD = DVDD5 = RVDD5=AVDD5B

24.3 DC Electrical Characteristics (2/2)

Ta= −40 to 85°C DVDD5 = RVDD5 = AVDD5B = 4.5V to 5.5V

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2) Gear 1/1	I_{CC}	fsys =80MHz	-	43.5	59.5	mA
IDLE (Note 2) Gear 1/1			-	15.5	25.5	
STOP		-	-	3.5	7.5	mA

Ta= −40 to 105°C DVDD5 = RVDD5 = AVDD5B = 4.5V to 5.5V

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2) Gear 1/1	I_{CC}	fsys =32MHz	-	23.5	36.5	mA
IDLE (Note 2) Gear 1/1			-	8.5	17.5	
STOP		-	-	3.5	7.5	mA

(Note 1) Ta = 25°C, DVDD5 = AVDD5B = RVDD5 = 5V, unless otherwise noted.

(Note 2) I_{CC} NORMAL, IDLE: All functions operates excluding A/D converter.

(Note 3) A/D reference voltage supply can not go into off state.

24.4 12-bit ADC Electrical Characteristics

DVDD5=RVDD5=AVDD5B/VREFHB = 4.5V to 5.5V,
DVSS=AVSSB/VREFLB=0V, Ta = -40 to 105°C

Parameter	Symbol	Rating	Min.	Typ.	Max.	Unit
Analog reference voltage (+)	VREFHA VREFHB	−		AVDD		V
Analog input voltage	VAIN	−	AVSS	−	AVDD	V
Analog supply current (Note 1)	IREF	DVSS = AVSS	−	3.5	4.5	mA
Supply current / A/D conversion	−	Except IREF (Note1)	−	−	5	mA
INL error	−	AIN resistance ≤600Ω AIN load capacitance ≦0.1μF Conversion time ≧2μs	−	−	±6	LSB
DNL error			−	−	±5	
Offset error			−	−	±6	
Full-scale error			−	−	±6	
Total error			−	−	-10 to +6	

(Note 1) A/D reference voltage supply can not go into off state.

(Note 2) 1LSB = (AVDD − AVSS) / 4096[V]

(Note 3) AVDD = AVDD5B, AVSS = AVSSB

(Note 4) The characteristic is measured under the condition in which the only ADC is operating.

24.5 AC Electrical Characteristics

AC measuremetn condition

Output levels: High 0.8VDD V/Low 0.2VDD, CL=30pF

Input levels: Refer to low-level input voltage and high-level input voltage in DC Electrical Characteristics.

In the table below, the letter x represents the period of the system clock (fsys). It varies depending on the programming of the clock gear function.

(Note 1) VDD = DVDD5 = RVDD5 = AVDD5B

24.5.1 Serial Channel Timing (SIO)

(1) I/O Interface mode

SCLK input mode   (VDD = 4.5V to 5.5V, Ta = −40 to 85°C<1 to 80MHz> / Ta = -40 to 105°C<1 to 32MHz>)

Parameter	Symbol	Equation		80MHz		Unit
		Min.	Max.	Min.	Max.	
SCLK Clock High width (input)	t_{SCH}	3x		37.5		ns
SCLK Clock Low width (input)	t_{SCL}	3x		37.5		ns
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$		75		ns
OutputData to SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 4x - 45$		0 (Note 2)		ns
OutputData hold or fall after SCLK rising (Note 1)	t_{OHS}	$t_{SYC}/2$		37.5		ns
Input Data valid SCLK rise or fall (Note 1)	t_{SRD}	30		30		ns
InputData hold or fall after SCLK rising (Note 1)	t_{HSR}	x + 30		42.5		ns

(Note 1) SCLK rise or fall:Measured relative to the programmed active edge of SCLK.
(Note 2) t_{oss} should be always positive. Therefore, set proper SCLK parameters (t_{SCY}, t_{SCL} and t_{SCY}) to keep t_{oss} positive. ($t_{oss} > 0$)

SCLK output mode (VDD = 4.5V to 5.5V, Ta = −40 to 85°C<1 to 80MHz> / Ta = -40 to 105°C<1 to 32MHz>)

Parameter	Symbol	Equation		80MHz		Unit
		Min.	Max.	Min.	Max.	
SCLK cycle (programmable)	t_{SCY}	4x		50		ns
OutputData to SCLK rise	t_{OSS}	$t_{scy}/2 - 20$		5		ns
OutputData hold after SCLK rising	t_{OHS}	$t_{scy}/2 - 20$		5		ns
InputData valid to SCLK rise	t_{SRD}	45		45		ns
InputData hold after SCLK rising	t_{HSR}	0		0		ns
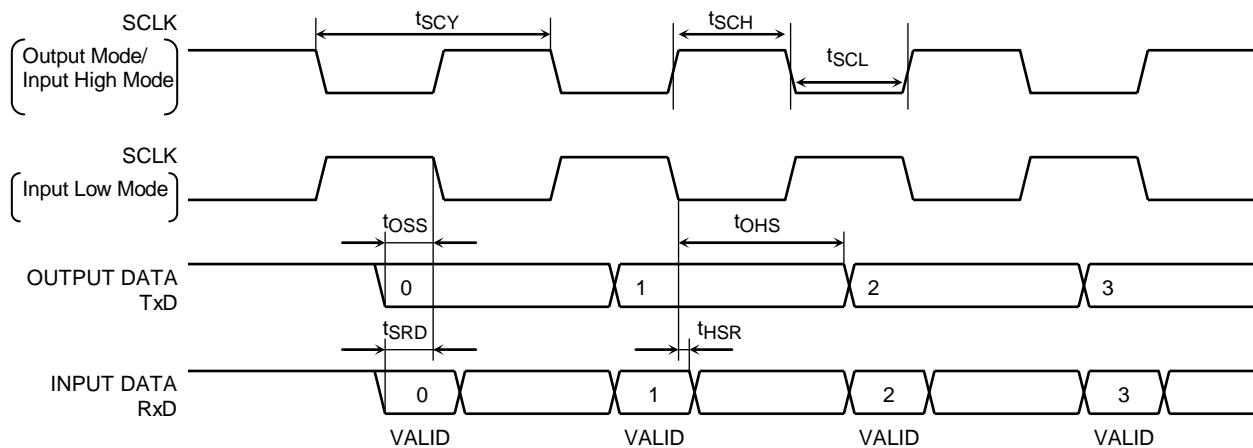
Fig 24-1 Serial channel timing(SIO)

24.5.2 Event Counter

Ta = −40 to 85°C(1 to 80MHz) / Ta = −40 to 105°C(1 to 32MHz)

Parameter	Symbol	Equation		80MHz		Unit
		Min.	Max.	Min.	Max.	
Clock low pulse width	t_{VCKL}	2x + 100		125		ns
Clock high pulse width	t_{VCKH}	2x + 100		125		ns

24.5.3 Capture

Ta = −40 to 85°C(1 to 80MHz) / Ta = −40 to 105°C(1 to 32MHz)

Parameter	Symbol	Equation		80MHz		Unit
		Min.	Max.	Min.	Max.	
Low pulse width	t_{CPL}	2x + 100		125		ns
High pulse width	t_{CPH}	2x + 100		125		ns

24.5.4 General Interrupts

Ta = −40 to 85°C(1 to 80MHz) / Ta = −40 to 105°C(1 to 32MHz)

Parameter	Symbol	Equation		80MHz		Unit
		Min.	Max.	Min.	Max.	
Low pulse width for INT0 to F	t_{INTAL}	x + 100		112.5		ns
High pulse width for INT0 to F	t_{INTAH}	x + 100		112.5		ns

24.5.5 STOP Release Interrupts

Ta = −40 to 105°C

Parameter	Symbol	Equation		80MHz		Unit
		Min.	Max.	Min.	Max.	
Low pulse width for INT0 to F	t_{INTBL}	100		100		ns
High pulse width for INT0 to F	t_{INTBH}	100		100		ns

24.5.6 Debug Communication

(1) SWD Interface

Parameter	Symbol	Min.	Max.	Unit
CLK cycle	Tdck	100	-	ns
DATA hold after CLK rising	Td1	4	-	ns
DATA valid after CLK rising	Td2	-	37	ns
DATA valid to CLK rising	Tds	20	-	ns
DATA hold after CLK falling	Tdh	15	-	ns

(2) JTAG Interface

Parameter	Symbol	Min.	Max.	Unit
CLK cycle	Tdck	100	-	ns
DATA hold after CLK falling	T_{d3}	4	-	ns
DATA valid after CLK falling	T_{d4}	-	37	ns
DATA valid to CLK rising	T_{ds}	20	-	ns
DATA hold after CLK rising	T_{dh}	15	-	ns
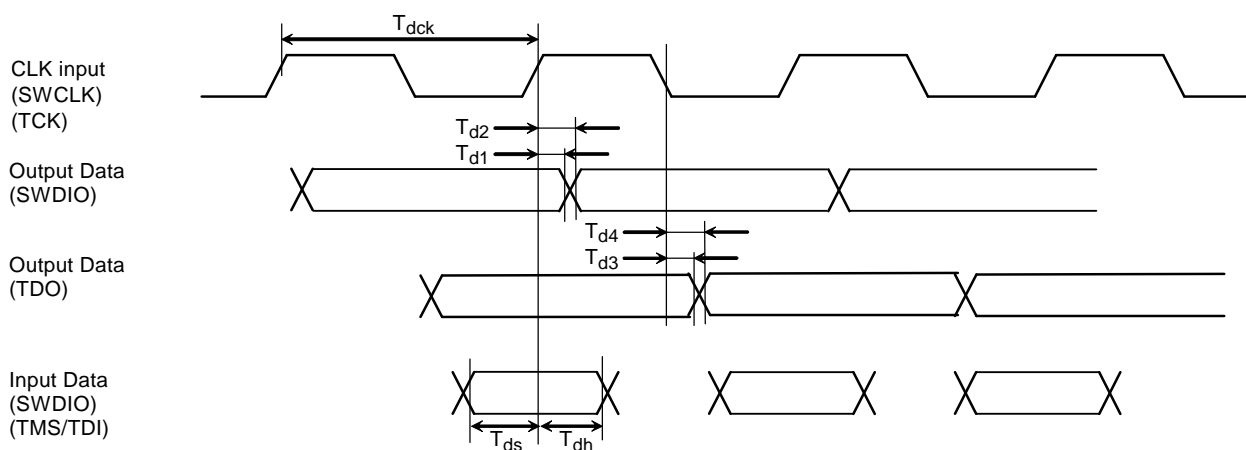
Fig 24-2 JTAG and SWD communication timing

24.5.7 TRACE Output

AC measuremetn condition

    Output levels: High 0.7VDD V/Low 0.3VDD

    Load capacitance: TRACECLK CL=25pF, TRACEDATA CL=20pF

Parameter	Symbol	Min.	Max.	Unit
TRACECLK cycle	t_{tclk}	25	-	ns
DATA valid after CLK rising	t_{setupr}	2	-	ns
DATA hold after CLK rising	t_{holdr}	1	-	ns
DATA valid after CLK falling	t_{setupf}	2	-	ns
DATA hold after CLK falling	t_{holdf}	1	-	ns
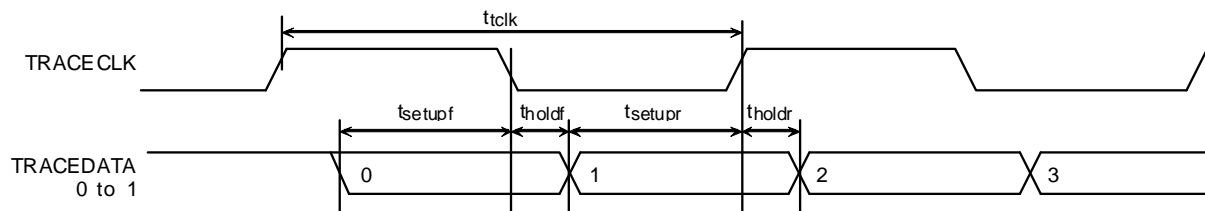
Fig 24-3 TRACE communication timing

24.6 Flash Characteristics

Ta = 0 to 70°C

Parameter	Rating	Min.	Typ.	Max.	Unit
Flash memory erase/write times				100	times

24.7 Built in oscillation characteristics

Ta= −40 to 105 °C

Parameter	Symbol	Min.	Typ.	Max.	Unit
Oscillation frequency	$f_{osc}2$	9	9.5	10	MHz

24.8 Oscillation Circuit

The TMPM372/3/4 has been evaluated by the oscillator vender below. Use this information when selecting external parts.

Note1: The load value of the oscillator is the sum of loads (C1 and C2) and the floating load of the actual assembled board. There is a possibility of operating error when using C1 and C2 values in the table below. When designing the board, design the minimum length pattern around the oscillator. We also recommend that oscillator evaluation be carried out using the actual board.

Note2: Do not be driven X1/X2 by external driver.

---

(1) Connection example
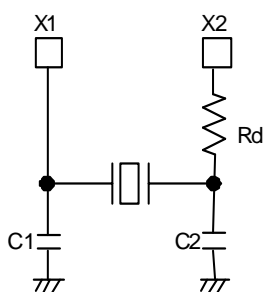
Fig 24-4 High-frequency oscillation connection

(2) Recommended ceramic oscillator

The TMPM372/3/4 recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.

Please refer to the following URL for details.

http://www.murata.co.jp

24.9 Handling Precaution

Solderability

Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230° C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming $\geqq$ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245° C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	

24.10 Note the power on

24.10.1 Using Power On Reset only

> **Note: When the power on reset is used without an external reset IC, DVDD5 and RVDD5 pins should reach the supply voltage (4.5 to 5.5V) within 3ms.**

Table 24.1 Warming-up time and Rising time of power line (POR only)

Symbol	Rating	Min	Typ.	Max	Unit
t_{PWUP}	Warming-up time after reset released		$2^{15}/f_{osc2}$		s
t_{DVDD}	Rising time of power line			3	ms

(Note) VDDs are DVDD5, RVDD5 and AVDD5B.
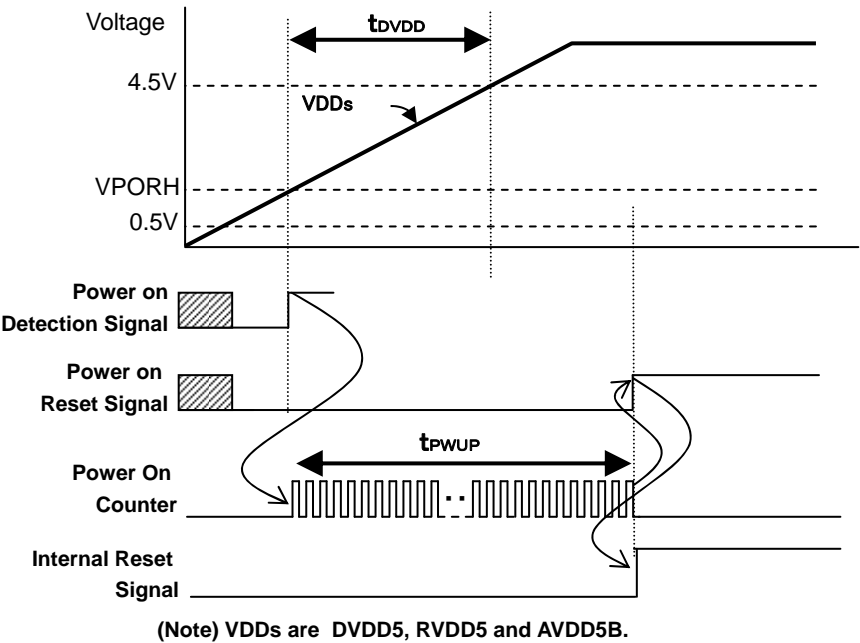
Fig 24.5 Power on Sequence(Using POR only)

24.10.2 Using External reset

(1) In case of the time of external reset shorter than POR

(Note) VDDs are DVDD5, RVDD5 and AVDD5B.
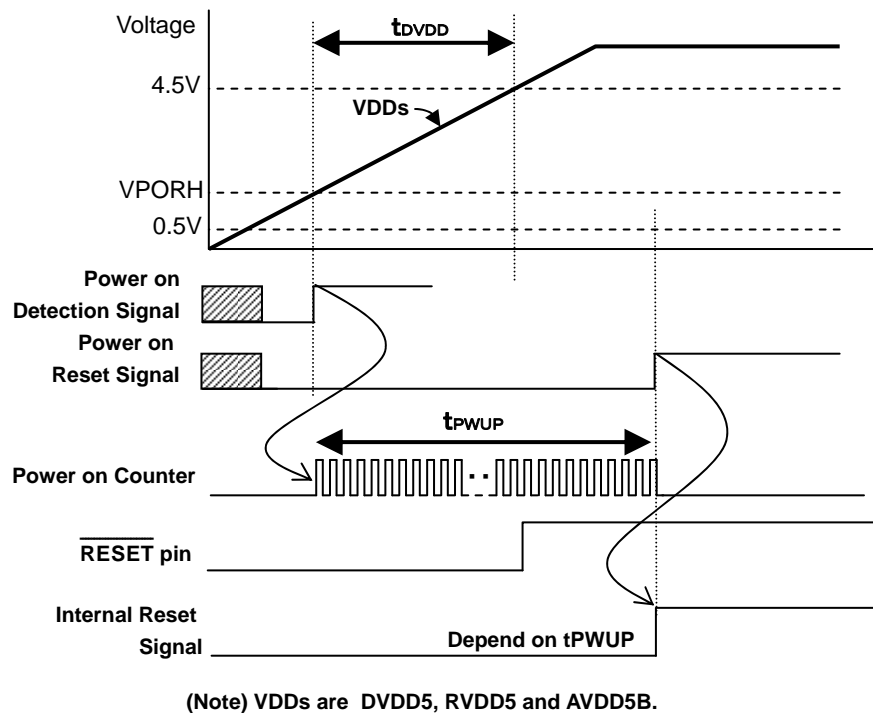
Fig 24.6 Power on Sequence(Using POR and External reset)(1)

(2) In case of the time of external reset longer than tPWUP

(Note) VDDs are DVDD5, RVDD5 and AVDD5B.
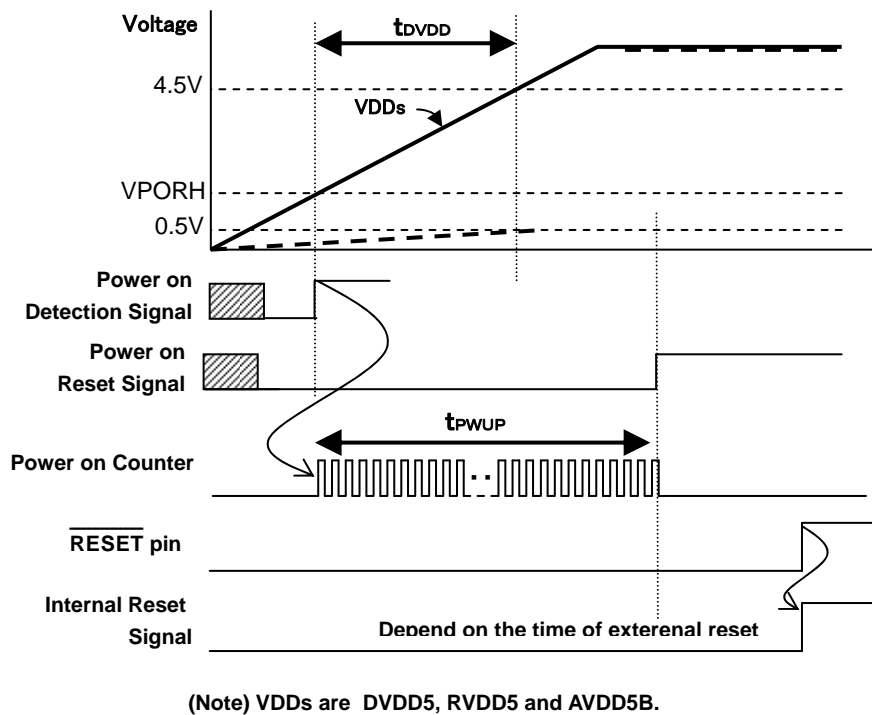
Fig 24.7 Power on Sequence(Using POR and External reset)(2)

(3) In case of the rising time of power line longer than tPWUP

(Note) In this case, must be reset from $\overline{\text{RESET}}$ pin.
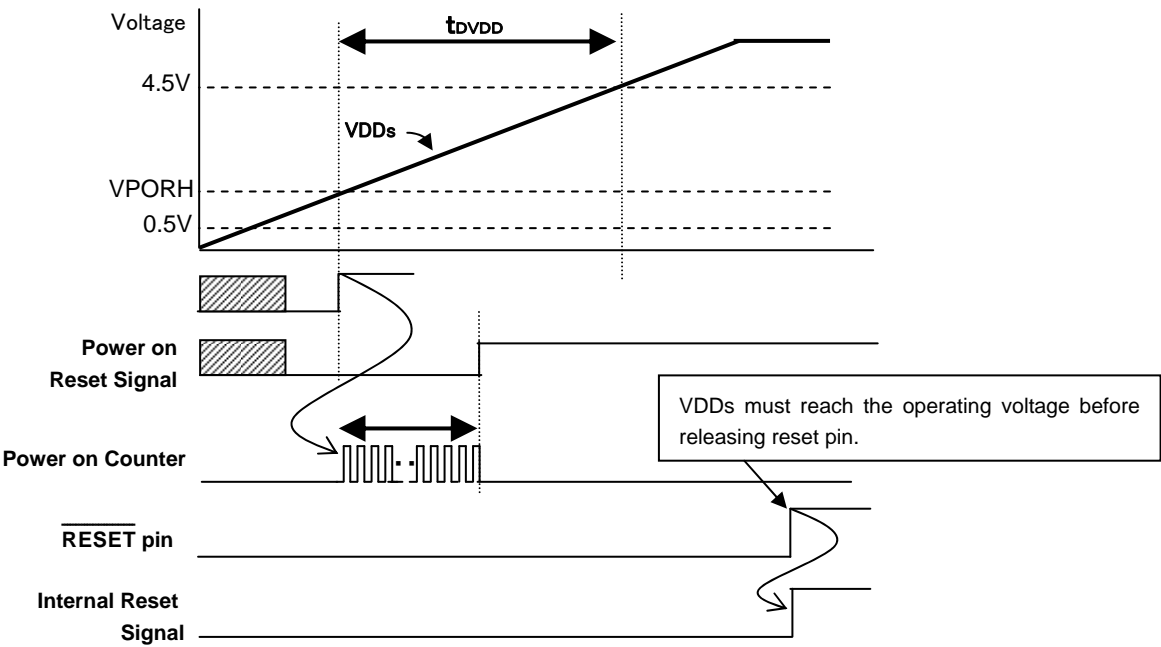
(Note) VDDs are DVDD5, RVDD5 and AVDD5B.

Fig 24.8 Power on Sequence (tDVDD > tPWUP)

25.   Package

Type : LQFP64-P-1010-0.50E

LQFP64-P-1010-0.50

Unit: mm
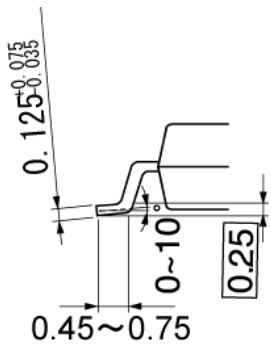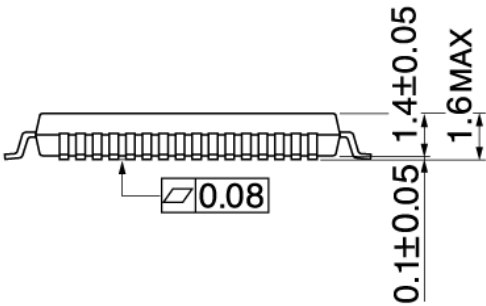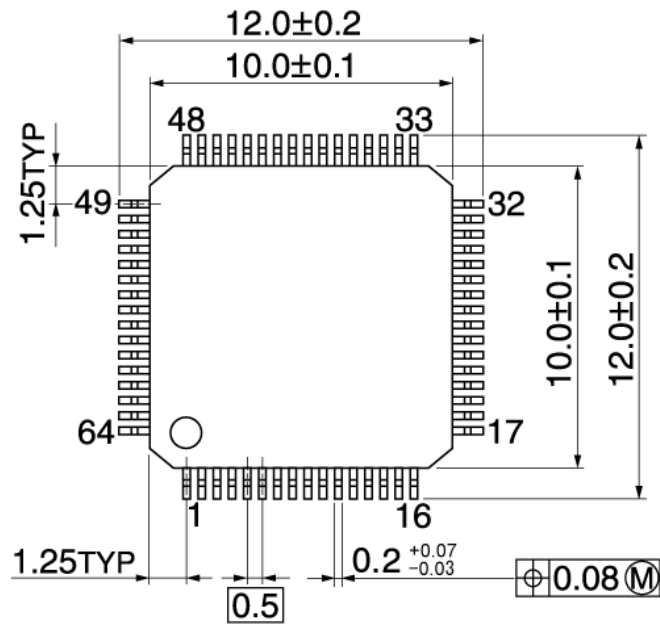
Type : LQFP48-P-0707-0.50C

LQFP48-P-0707-0.50C

Unit: mm
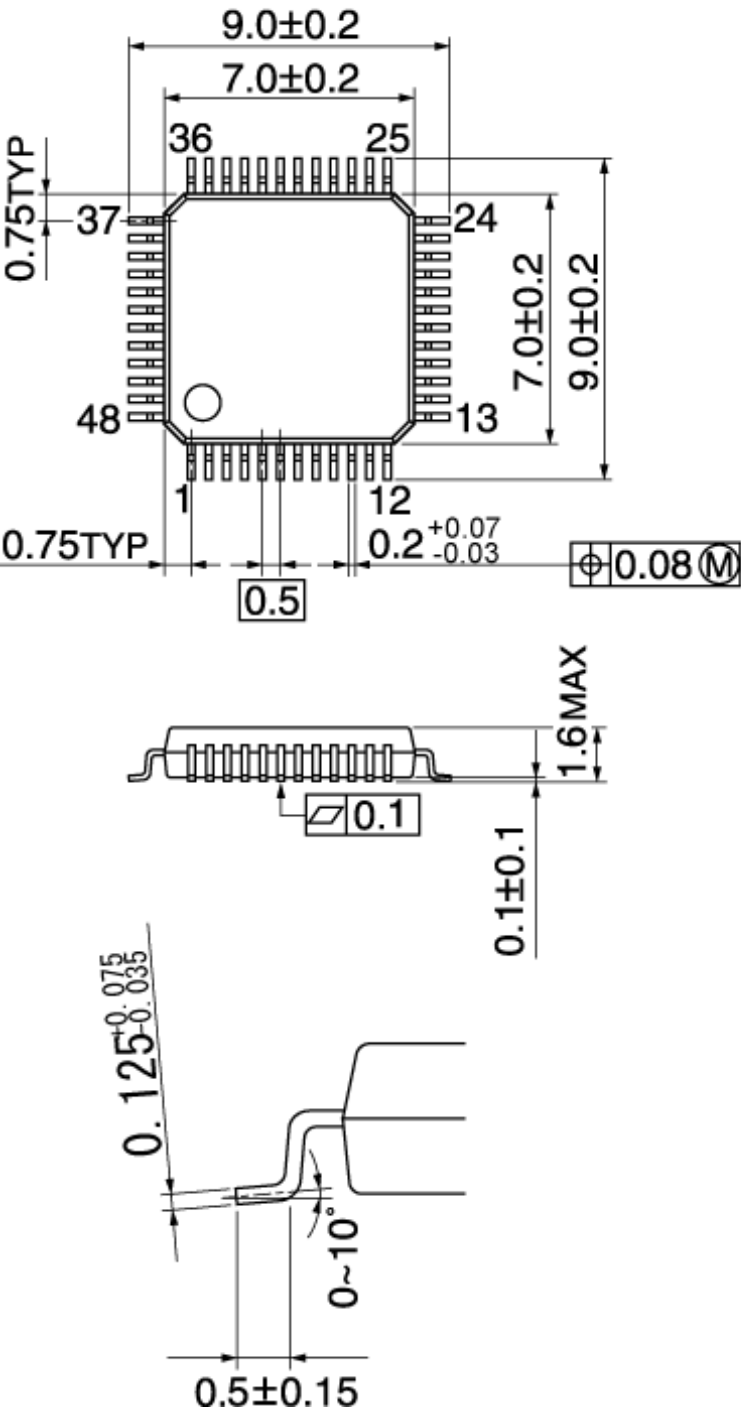
Type : LQFP44-P-1010-0.80A

LQFP44-P-1010-0.80

Unit: mm
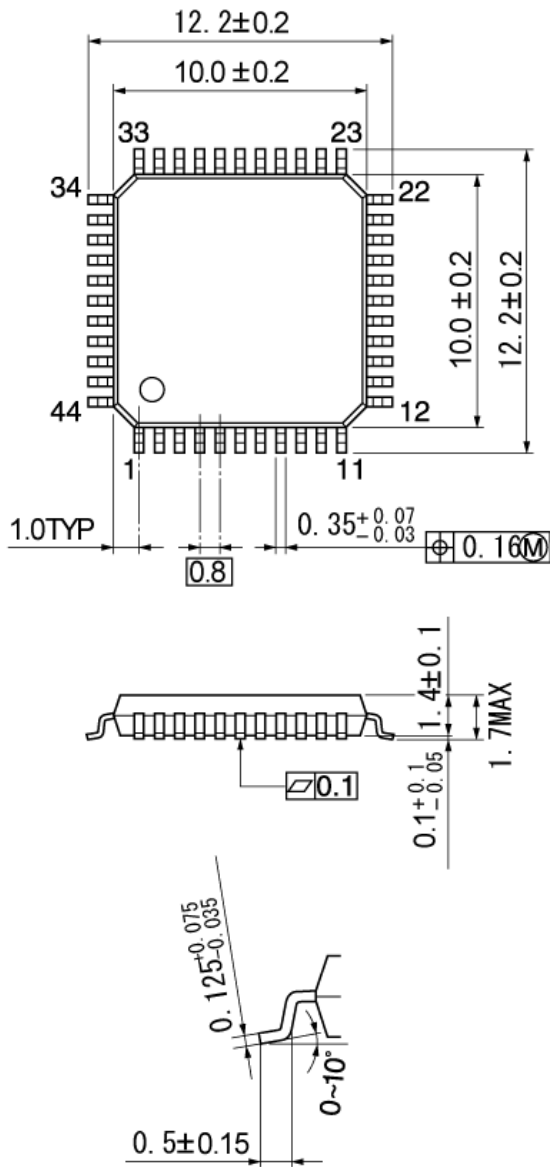
TOSHIBA

RESTRICTIONS ON PRODUCT USE