

**TOSHIBA**

**32-bit RISC Microcontroller  
TX03 Series**

**TMPM370FYDFG/FYFG**

**TOSHIBA CORPORATION**

Semiconductor Company

## Revision History

| Date      | Revision |               |
|-----------|----------|---------------|
| 2010/9/24 | Rev 1    | First Release |

\*\*\*\*\*  
ARM, ARM Powered, AMBA, ADK, ARM9TDMI, TDMI, PrimeCell, RealView, Thumb, Cortex, Coresight, ARM9, ARM926EJ-S, Embedded Trace Macrocell, ETM, AHB, APB, and KEIL are registered trademarks or trademarks of ARM Limited in the EU and other countries.  
\*\*\*\*\*



32-bit RISC microcontroller TX03 series

# TMPM370FYDFG

# TMPM370FYFG

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

| Product No.  | On chip Flash ROM | On chip RAM | Package              |
|--------------|-------------------|-------------|----------------------|
| TMPM370FYDFG | 256 Kbyte         | 10 Kbyte    | QFP100-P-1420-0.65Q  |
| TMPM370FYFG  | 256 Kbyte         | 10 Kbyte    | LQFP100-P-1414-0.50H |

## 1.1 Features

### (1) ARM Cortex-M3 microcontroller core

- 1) 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.
- 2) Both high performance and low power consumption have been achieved
  - High performance
    - A 32-bit multiplication (32×32=32 bit) can be executed with one clock
    - Division takes between 2 and 12 cycles depending on dividend and divisor
  - Low power consumption
    - Optimized design using a low power consumption library
    - Standby function that stops the operation of the microcontroller core
- 3) High-speed interrupt response suitable for real-time control
  - An interruptible long instruction
  - Stack push automatically handled by hardware

### (2) Interrupt source

- Internal: 62 factors...The order of precedence can be set over 7 levels (except the watchdog timer interrupt).
- External: 16 factors...The order of precedence can be set over 7 levels

### (3) Input/ output ports

- Input/Output: 74 pins    Input: 2pins

### (4) Watchdog timer(WDT): 1 channel

- 26 cycles of binary counter

### (5) Power\_On reset function(POR)

### (6) Voltage detect function(VLTD)

### (7) Oscillation frequency detect function(OFD)

- ( 8 ) Vector engine(VE): 1 unit
  - Calculation circuit for motor control
  - corresponding to 2 motors
- ( 9 ) Programmable motor driver(PMD): 2channel
  - 3phase complementary PWM generator
  - Synchronous A/D convert start trigger generator
  - Emergency protective function(EMG pin/comparator output)
- ( 10 ) Encoder input circuit(ENC): 2 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
- ( 11 ) 16-bit timer(TMRB): 8 channel
  - 16-bit interval timer mode
  - 16-bit event counter mode
  - 16-bit PPG output
  - External trigger PPG output
  - Input capture function
- ( 12 ) General-purpose serial interface(SIO): 4 channel
  - Either UART mode or synchronous mode can be selected (4byte FIFO equipped)
- ( 13 ) 12-bit A/D converter (ADC): 2unit( 22 channel for analog input)
  - Start by the internal trigger: TMRB interrupt / PMD trigger
  - Constant conversion mode
  - AD conversion monitoring function(2ch)
  - Conversion speed 2usec (@ ADC conversion clock = 40MHz)
- ( 14 ) OP-Amp(AMP): 4 channel
  - 8 gain can be selected
- ( 15 ) comparator(CMP): 3+1 channel
  - Protection for motor emergence stop
  - 2 input type(OP-Amp output/analog input)
- ( 16 ) Standby mode
  - Standby mode: IDLE, STOP
- ( 17 ) 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
- ( 18 ) Endian
  - Little endian
- ( 19 ) Maximum operating frequency
  - 80MHz
- ( 20 ) Operating voltage range
  - 4.5V~5.5V (with on-chip regulator)
- ( 21 ) Temperature range
  - -40°C~85°C (except during Flash writing/ erasing and debugging)
  - 0°C~70°C (during Flash writing/ erasing and debugging)

1.2 Block Diagram

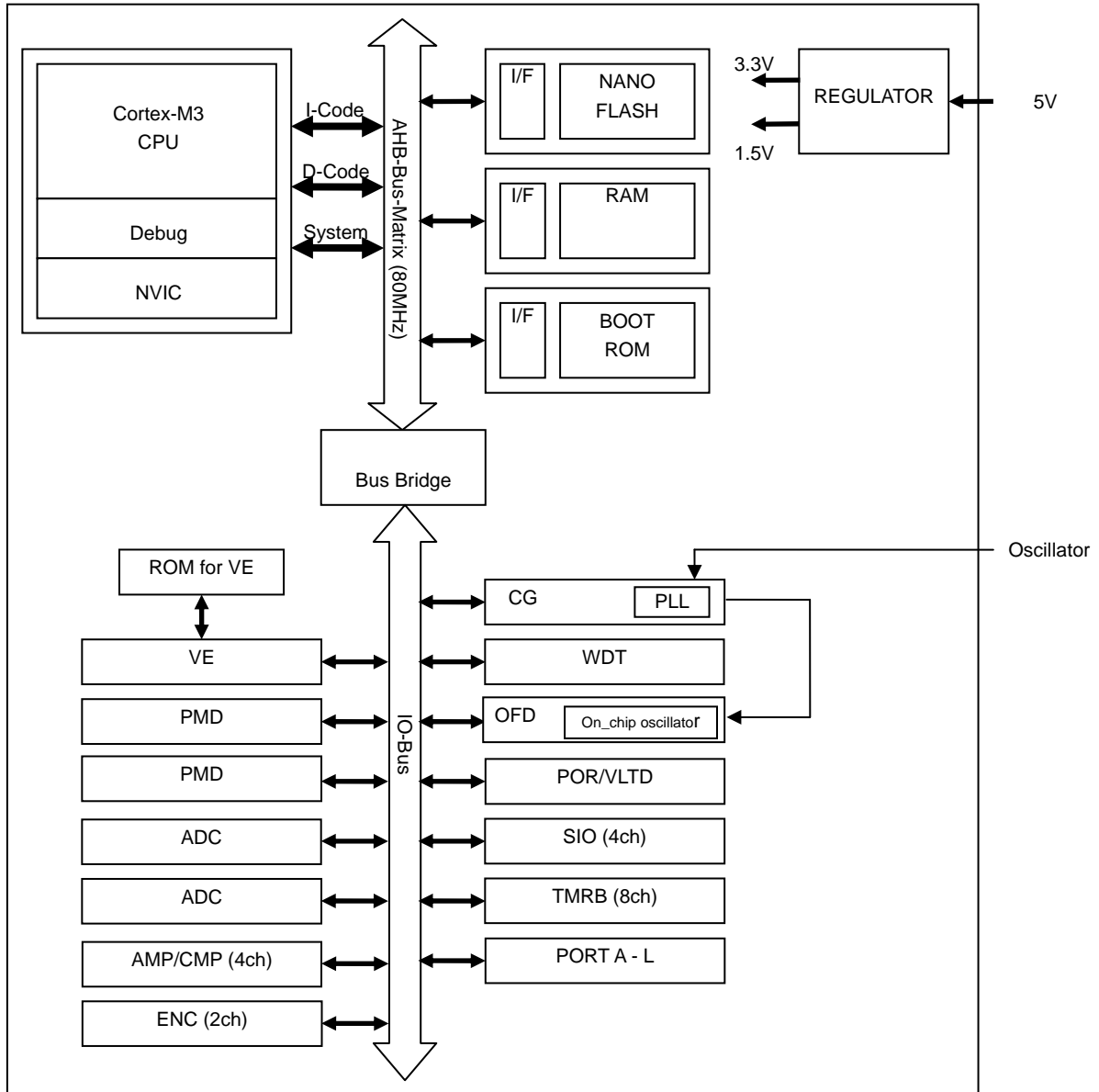


Fig1.1 TMPM370FYDFG/ FYFG block diagram

## 2. Pin Layout and Pin Functions

This chapter describes the pin layout, pin names and pin functions of TMPM370FYFG/TMPM370FYDFG.

### 2.1 Pin Layout (Top view)

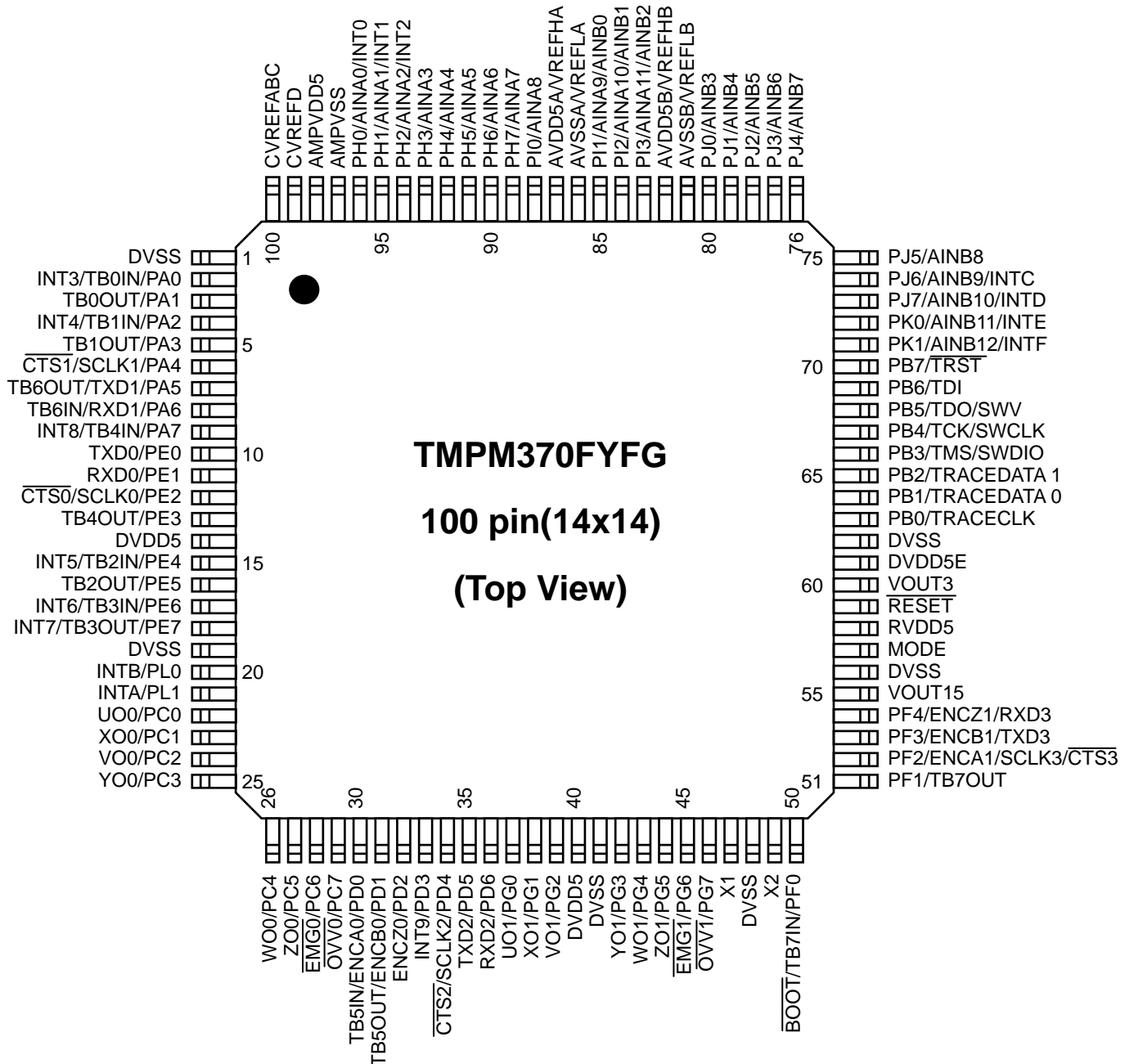


Fig 2-1 Pin layout (TMPM370FYFG)

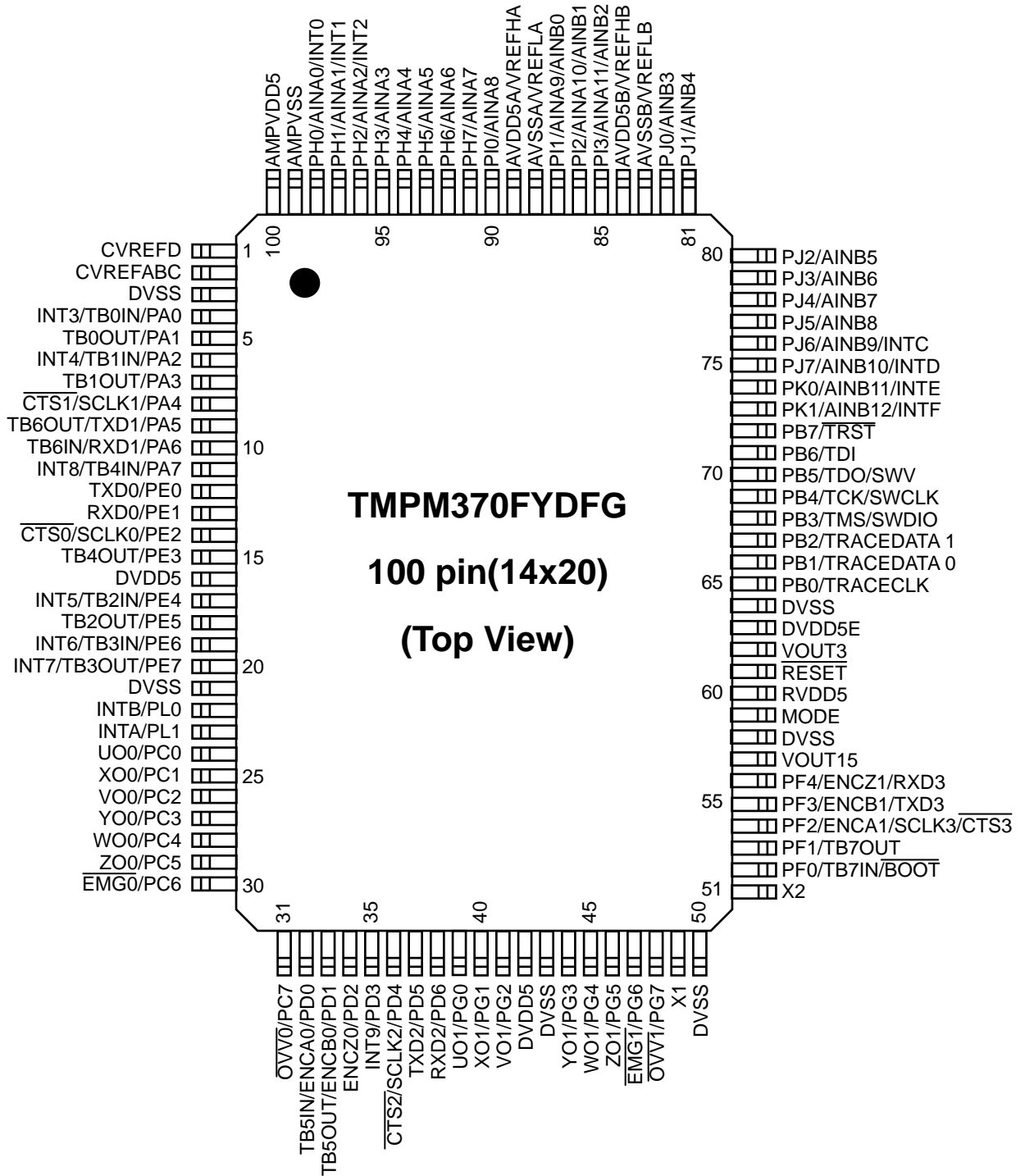


Fig 2-2 Pin layout (TMPM370FYDFG)



## 2.2 Pin function

Table 2.1 shows the pin functions of TMPM370FYFG/FYDFG. 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 (1/5)

| Pin name                                 | Output during Reset | SCHMITT<br>(O:Yes) | Open Drain<br>mode | High<br>Speed |
|--|---------------------|--------------------|--------------------|---------------|
| CVREFD                                   | Hi-Z                |                    |                    |               |
| CVREFABC                                 | Hi-Z                |                    |                    |               |
| DVSS                                     |                     |                    |                    |               |
| PA0<br>TB0IN/INT3                        | Hi-Z                | O                  | O                  |               |
| PA1<br>TB0OUT                            | Hi-Z                | O                  | O                  |               |
| PA2<br>TB1IN<br>INT4                     | Hi-Z                | O                  | O                  |               |
| PA3<br>TB1OUT                            | Hi-Z                | O                  | O                  |               |
| PA4<br>SCLK1<br>$\overline{\text{CTS1}}$ | Hi-Z                | O                  | O                  |               |
| PA5<br>TXD1<br>TB6OUT                    | Hi-Z                | O                  | O                  |               |
| PA6<br>RXD1<br>TB6IN                     | Hi-Z                | O                  | O                  |               |
| PA7<br>TB4IN<br>INT8                     | Hi-Z                | O                  | O                  |               |
| PE0<br>TXD0                              | Hi-Z                | O                  | O                  |               |
| PE1<br>RXD0                              | Hi-Z                | O                  | O                  |               |
| PE2<br>SCLK0<br>$\overline{\text{CTS0}}$ | Hi-Z                | O                  | O                  |               |
| PE3<br>TB4OUT                            | Hi-Z                | O                  | O                  |               |
| DVDD5                                    |                     |                    |                    |               |
| PE4<br>TB2IN<br>INT5                     | Hi-Z                | O                  | O                  |               |
| PE5<br>TB2OUT                            | Hi-Z                | O                  | O                  |               |
| PE6<br>TB3IN<br>INT6                     | Hi-Z                | O                  | O                  |               |
| PE7<br>TB3OUT<br>INT7                    | Hi-Z                | O                  | O                  |               |
| DVSS                                     |                     |                    |                    |               |

Table 2.1 Pin functions (2/5)

| Pin name                                 | Output during Reset | SCHMITT<br>(O:Yes) | Open Drain<br>mode | High<br>Speed |
|--|---------------------|--------------------|--------------------|---------------|
| PL0<br>INTB                              | Hi-Z                | ○                  |                    |               |
| PL1<br>INTA                              | Hi-Z                | ○                  |                    |               |
| PC0<br>UO0                               | Hi-Z                | ○                  | ○                  |               |
| PC1<br>XO0                               | Hi-Z                | ○                  | ○                  |               |
| PC2<br>VO0                               | Hi-Z                | ○                  | ○                  |               |
| PC3<br>YO0                               | Hi-Z                | ○                  | ○                  |               |
| PC4<br>WO0                               | Hi-Z                | ○                  | ○                  |               |
| PC5<br>ZO0                               | Hi-Z                | ○                  | ○                  |               |
| PC6<br>$\overline{\text{EMG0}}$          | Hi-Z                | ○                  | ○                  |               |
| PC7<br>$\overline{\text{OVV0}}$          | Hi-Z                | ○                  | ○                  |               |
| PD0<br>ENCA0<br>TB5IN                    | Hi-Z                | ○                  | ○                  |               |
| PD1<br>ENCB0<br>TB5OUT                   | Hi-Z                | ○                  | ○                  |               |
| PD2<br>ENCZ0                             | Hi-Z                | ○                  | ○                  |               |
| PD3<br>INT9                              | Hi-Z                | ○                  | ○                  |               |
| PD4<br>SCLK2<br>$\overline{\text{CTS2}}$ | Hi-Z                | ○                  | ○                  |               |
| PD5<br>TXD2                              | Hi-Z                | ○                  | ○                  |               |
| PD6<br>RXD2                              | Hi-Z                | ○                  | ○                  |               |
| PG0<br>UO1                               | Hi-Z                | ○                  | ○                  |               |
| PG1<br>XO1                               | Hi-Z                | ○                  | ○                  |               |
| PG2<br>VO1                               | Hi-Z                | ○                  | ○                  |               |
| DVDD5                                    |                     |                    |                    |               |
| DVSS                                     |                     |                    |                    |               |

Table 2.1 Pin functions (3/5)

| Pin name                      | Output during Reset | SCHIMITT<br>(O:Yes) | Open Drain<br>mode | High<br>Speed |
|-------------------------------|---------------------|---------------------|--------------------|---------------|
| PG3<br>YO1                    | Hi-Z                | O                   | O                  |               |
| PG4<br>WO1                    | Hi-Z                | O                   | O                  |               |
| PG5<br>ZO1                    | Hi-Z                | O                   | O                  |               |
| PG6<br>EMG1                   | Hi-Z                | O                   | O                  |               |
| PG7<br>OVV1                   | Hi-Z                | O                   | O                  |               |
| X1                            |                     |                     |                    |               |
| DVSS                          | Hi-Z                | O                   |                    |               |
| X2                            |                     |                     |                    |               |
| PF0<br>TB7IN<br>BOOT          | Pull UP             | O                   | O                  |               |
| PF1<br>TB7OUT                 | Hi-Z                | O                   | O                  |               |
| PF2<br>ENCA1<br>SCLK3<br>CTS3 | Hi-Z                | O                   | O                  |               |
| PF3<br>ENCB1<br>TXD3          | Hi-Z                | O                   | O                  |               |
| PF4<br>ENCZ1<br>RXD3          | Hi-Z                | O                   | O                  |               |
| VOUT15                        |                     |                     |                    |               |
| DVSS                          |                     |                     |                    |               |
| MODE                          | Hi-Z                | O                   |                    |               |
| RVDD5                         |                     |                     |                    |               |
| RESET                         | Pull UP             | O                   |                    |               |
| VOUT3                         |                     |                     |                    |               |
| DVDD5E                        |                     |                     |                    |               |
| DVSS                          |                     |                     |                    |               |
| PB0<br>TRACECLK               | Hi-Z                | O                   |                    | O             |
| PB1<br>TRACEDATA 0            | Hi-Z                | O                   |                    | O             |
| PB2<br>TRACEDATA 1            | Hi-Z                | O                   |                    | O             |
| PB3<br>TMS<br>SWDIO           | Pull UP             | O                   |                    | O             |
| PB4<br>TCK<br>SWCLK           | Pull DOWN           | O                   |                    | O             |

Table 2.1 Pin functions (4/5)

| Pin name                        | Output during Reset | SCHIMITT<br>(O:Yes) | Open Drain<br>mode | High<br>Speed |
|---------------------------------|---------------------|---------------------|--------------------|---------------|
| PB5<br>TDO<br>SWV               | Hi-Z                | O                   |                    | O             |
| PB6<br>TDI                      | Pull UP             | O                   |                    |               |
| PB7<br>$\overline{\text{TRST}}$ | Pull UP             | O                   |                    |               |
| PK1<br>AINB12<br>INTF           | Hi-Z                | O                   | O                  |               |
| PK0<br>AINB11<br>INTE           | Hi-Z                | O                   | O                  |               |
| PJ7<br>AINB10<br>INTD           | Hi-Z                | O                   | O                  |               |
| PJ6<br>AINB9<br>INTC            | Hi-Z                | O                   | O                  |               |
| PJ5<br>AINB8                    | Hi-Z                | O                   | O                  |               |
| PJ4<br>AINB7                    | Hi-Z                | O                   | O                  |               |
| PJ3<br>AINB6                    | Hi-Z                | O                   | O                  |               |
| PJ2<br>AINB5                    | Hi-Z                | O                   | O                  |               |
| PJ1<br>AINB4                    | Hi-Z                | O                   | O                  |               |
| PJ0<br>AINB3                    | Hi-Z                | O                   | O                  |               |
| AVSSB<br>VREFLB                 |                     |                     |                    |               |
| AVDD5B<br>VREFHB                |                     |                     |                    |               |
| PI3<br>AINA11<br>AINB2          | Hi-Z                | O                   | O                  |               |
| PI2<br>AINA10<br>AINB1          | Hi-Z                | O                   | O                  |               |
| PI1<br>AINA9<br>AINB0           | Hi-Z                | O                   | O                  |               |
| AVSSA<br>VREFLA                 |                     |                     |                    |               |
| AVDD5A<br>VREFHA                |                     |                     |                    |               |
| PI0<br>AINA8                    | Hi-Z                | O                   | O                  |               |

Table 2.1 Pin function (5/5)

| Pin name             | Output during Reset | SCHMITT<br>(O:Yes) | Open Drain<br>mode | High<br>Speed |
|----------------------|---------------------|--------------------|--------------------|---------------|
| PH7<br>AINA7         | Hi-Z                | O                  | O                  |               |
| PH6<br>AINA6         | Hi-Z                | O                  | O                  |               |
| PH5<br>AINA5         | Hi-Z                | O                  | O                  |               |
| PH4<br>AINA4         | Hi-Z                | O                  | O                  |               |
| PH3<br>AINA3         | Hi-Z                | O                  | O                  |               |
| PH2<br>AINA2<br>INT2 | Hi-Z                | O                  | O                  |               |
| PH1<br>AINA1<br>INT1 | Hi-Z                | O                  | O                  |               |
| PH0<br>AINA0<br>INT0 | Hi-Z                | O                  | O                  |               |
| AMPVSS               |                     |                    |                    |               |
| AMPVDD5              |                     |                    |                    |               |

Table 2.2 Operating voltage of each Pin

| Pin name                  | Operating voltage            |  |
|---------------------------|------------------------------|--|
| X1,X2                     | Internal 1.5V/CVSS           | Do not be driven by external circuit.                                |
| $\overline{\text{RESET}}$ | DVDD5/DVSS                   |  |
| MODE                      | DVDD5/DVSS                   | must be connect with GND   |
| CVREFABC, CVREFD          | AMPVDD5/AMPVSS               | Compare(reference) voltage input for comparator ABC and comparator D |
| I/O<br>(PAx,PCx-PGx,PLx)  | DVDD5/DVSS                   |  |
| I/O(PBx)                  | DVDD5E/DVSS                  |  |
| AIN(PHx-PKx)              | AVDD5A/AVSSB<br>AVDD5B/AVSSB |  |

Table 2.3 voltage range of each pin

| Pin name      | Voltage range |   |
|---------------|---------------|---|
| DVDD5         | 4.5 ~ 5.5V    | For I/O ports   |
| RVDD5         |               | For internal circuit  |
| AVDD5A,AVDD5B |               | For ADCs  |
| AMPVDD5       |               | For operational amp and comparator  |
| DVDD5E        |               | DVDD5E must be connect with DVDD5   |
| VOUT15        | 1.35 ~ 1.65V  | VOUT15 must be connected with DVSS through 3.3~4.7uF capacitor for supply power to internal circuit.. (Note)<br>Do not supply power to external circuits. |
| VOUT3         | 2.7 ~ 3.6V    | VOUT3 must be connected with DVSS through 3.3~4.7uF capacitor for supply power to internal circuit.. (Note)<br>Do not supply power to external circuits.  |
| DVSS          | GND           |   |
| AVSSA,AVSSB   |               |   |
| AMPVSS        |               |   |

Note: 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 TMPM370FY. For further information on each revision, see the documents issued by ARM Limited.

| Product Name | Core Revision |
|--------------|---------------|
| TMPM370FY    | r2p0-00rel0   |

### 3.2 Optional block

The Cortex-M3 core has the optional blocks. The optional blocks of the revision r2p0 are below;

| Optional block                 | Implementation |
|--------------------------------|----------------|
| FPB                            | O              |
| DWT                            | O              |
| ITM                            | O              |
| MPU                            | X              |
| ETM <sup>TM</sup>              | O              |
| AHB-AP                         | O              |
| AHB trace macro-cell interface | O              |
| TPIU                           | O              |
| WIC                            | X              |

O: Implement

X: Not implement

### 3.3 Event

TMPM370FY is not support event input/output. Do not use SEV instruction and WFE instruction.

### 3.4 SLEEPDEEP

TMPM370FY is not support SLEEPDEEP. Do not set SLEEPDEEP bit.

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

### 3.5 Exclusive access

TMPM370FY is not 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

TMPM370 has Power-on reset circuit, power-on reset signal is generated when power supply is turned on. More information is described in Chapter "Power-on Reset Circuit (POR)" and "Note the power on of "Electrical Characteristics".

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 TMPM370, 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. High-speed oscillator is started oscillation. Note that the PLL multiplication circuit stops after releasing the reset. Therefore, set CGOSCCR register and CGPLLSEL register to use PLL multiplication circuit. Regarding detail information, see to "Clock/Mode control".

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". Detail information is described in "Attention when power supply is turned on" of "Electrical Characteristics".

(Note 2) The reset operation may alter the internal RAM state.

## 4 Dubug Interface

### 4.1 Specification Overview

The TM370 contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the Debug Interface and the Embedded Trace Macrocell™ (ETM) unit for trace output. Trace data is output to the dedicated pins (TRACEDATA[0]-[1], SWV) via the on-chip Trace Port Interface Unit (TPIU).

For details about SWJ-DP, ETM and TPIU, refer to “Cortex-M3 Technical Reference Manual”.

### 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, TRST).

### 4.3 Features of ETM

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

### 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<br>Pin name | Name of<br>port | JTAG debug function |                          | SW debug                |                               |
|--------------------|-----------------|---------------------|--------------------------|-------------------------|-------------------------------|
|                    |                 | I/O                 | Discription              | I/O                     | Discription                   |
| TMS/SWDIO          | PB3             | Input               | JTAG Test Mode Selection | I/O                     | Serial Wire Data Input/Output |
| TCK/SWCLK          | PB4             | Input               | JTAG Test Check          | Input                   | Serial Wire Clock             |
| TDO/SWV            | PB5             | Output              | JTAG Test Data Output    | -<br>(Output)<br>(Note) | (Serial Wire Viewer Output)   |
| TDI                | PB6             | Input               | JTAG Test Data Input     | -                       | -                             |
| TRST               | PB7             | Input               | JTAG Test RESET          | -                       | -                             |
| TRACECLK           | PB0             | Output              | TRACE Clock Output       |                         |                               |
| TRACEDATA0         | PB1             | Output              | TRACE DATA Output0       |                         |                               |
| TRACEDATA1         | PB2             | Output              | TRACE DATA 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.

The 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 | Port Setting After Reset |              |               |                 |                   |
|-----------------|-----------------|----------------|--------------------------|--------------|---------------|-----------------|-------------------|
|                 |                 |                | Function (PBFR)          | Input (PBIE) | Output (PBCR) | Pull-up (PBPUP) | Pull-down (PBPDN) |
| PORT            | PB0             | TRACECLK       | 0                        | 0            | 0             | 0               | -                 |
| PORT            | PB1             | TARCEATA0      | 0                        | 0            | 0             | 0               | -                 |
| PORT            | PB2             | TRACEDATA1     | 0                        | 0            | 0             | 0               | -                 |
| DEBUG           | PB3             | TMS/SWDIO      | 1                        | 1            | 1             | 1               | -                 |
| DEBUG           | PB4             | TCK/SWCLK      | 1                        | 1            | 0             | -               | 1                 |
| DEBUG           | PB5             | TDO/SWV        | 1                        | 0            | 1             | 0               | -                 |
| DEBUG           | PB6             | TDI            | 1                        | 1            | 0             | 1               | -                 |
| DEBUG           | PB7             | TRST           | 1                        | 1            | 0             | 1               | -                 |

- ; There is no register.

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

### 4.6 How to connect

For how to connect a debug tool, refer to the method recommended by each manufacturer.

Debug interface pins have pull-up or pull-down register. When connect with pull-up or pull-down registers ,be sure their settings.

### 4.7 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 settings.

Table 4-3 Debug Interface

| Usag                   | Using Debug Interface pins |     |             |               |               |                |                |              |
|------------------------|----------------------------|-----|-------------|---------------|---------------|----------------|----------------|--------------|
|                        | TRST                       | TDI | TDO/<br>SWV | TCK/<br>SWCLK | TMS/<br>SWDIO | TRACE<br>DATA1 | TARCE<br>DATA0 | TRACE<br>CLK |
| JTAG+SW (After RESET)  | ○                          | ○   | ○           | ○             | ○             | ×              | ×              | ×            |
| JTAG+SW (No TRST)      | ×                          | ○   | ○           | ○             | ○             | ×              | ×              | ×            |
| JTAG+TRACE             | ○                          | ○   | ○           | ○             | ○             | ○              | ○              | ○            |
| SW                     | ×                          | ×   | ×           | ○             | ○             | ×              | ×              | ×            |
| SW+SWV                 | ×                          | ×   | ○           | ○             | ○             | ×              | ×              | ×            |
| Disable Debug function | ×                          | ×   | ×           | ×             | ×             | ×              | ×              | ×            |

○ : Enable、 × : Disable (Can use general purpose port)

#### 4.8 Peripherals operation during HALT mode (one time stop of running program)

When Break during debugging, Cortex-M3 CPU core going into HALT mode. Watch dog timer (WDT) is stopped counting automatically. Other peripherals are continue operating.

## 5 Memory Map

The memory maps for the TMPM370FY are based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and internal I/O regions of the TMPM370FY 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.

See "Special Function Registers" for details on the internal I/O region.

### 5.1 Memory Map

Fig 5-1 shows the memory map of the TMPM370FY.

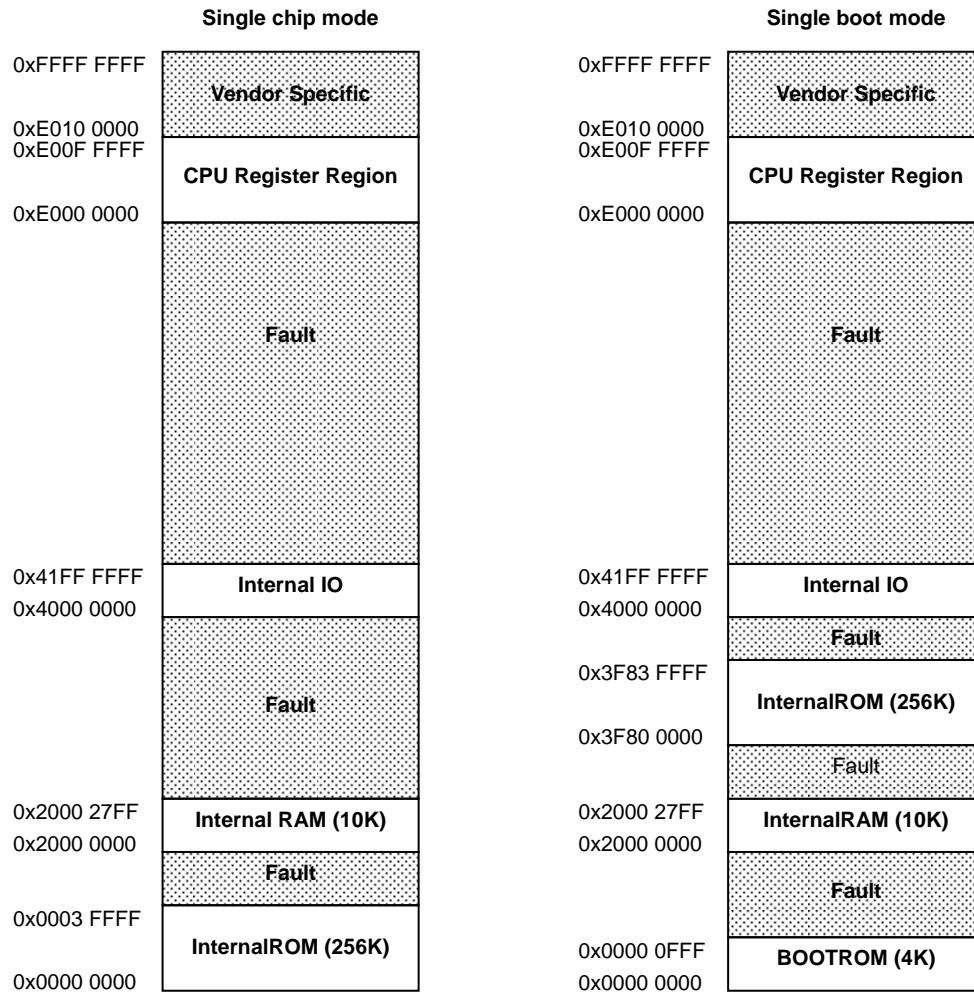


Fig 5-1 Memory Map

## 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 TMPM370FY 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)

|             |              |           |           |                                     |              |   |           |            |     |
|-------------|--------------|-----------|-----------|-------------------------------------|--------------|---|-----------|------------|-----|
|             | <b>7</b>     | <b>6</b>  | <b>5</b>  | <b>4</b>                            | <b>3</b>     | <b>2</b>  | <b>1</b>  | <b>0</b>   |     |
| 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<br>000: fc                   100: fc/2<br>001: reserved           101: fc/4<br>010: reserved           110: fc/8<br>011: reserved           111: fc/16 |           |            |     |
|             | <b>15</b>    | <b>14</b> | <b>13</b> | <b>12</b>                           | <b>11</b>    | <b>10</b>   | <b>9</b>  | <b>8</b>   |     |
| 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<br>clock<br>0:fgear<br>1:fc | "0" is read. | Prescaler clock<br>000: fperiph           100: fperiph/16<br>001: fperiph/2       101: fperiph/32<br>010: fperiph/4       110: Reserved<br>011: fperiph/8       111: Reserved     |           |            |     |
|             | <b>23</b>    | <b>22</b> | <b>21</b> | <b>20</b>                           | <b>19</b>    | <b>18</b>   | <b>17</b> | <b>16</b>  |     |
| 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" |     |
|             | <b>31</b>    | <b>30</b> | <b>29</b> | <b>28</b>                           | <b>27</b>    | <b>26</b>   | <b>25</b> | <b>24</b>  |     |
| 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.





## 6.2.2.3 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<br>000: Reserved<br>001: STOP<br>010: Reserved<br>011: IDLE<br>100: Reserved<br>101: Reserved<br>110: Reserved<br>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<br>Write "1". |
|             | 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<br>0: Active<br>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.

<Bit 8><RXEN> : Write "1".

<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<br>0: fosc<br>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.

## 6.2.2.5 System Clock Selection Register (CGCKSEL)

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

- fosc : Clock input from high-speed oscillator (X1 and X2)
- 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>
- $\Phi T0$  : Clock specified by CGSYSCR<PRCK2:0> (prescaler clock)

The high-speed clock gear (fgear) and the prescaler clock  $\Phi T0$  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 : 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 fosc.

- fc = fosc
- fsys = fc (=fosc)
- fperiph = fc (=fosc)
- $\Phi T0$  = fperiph (=fosc)

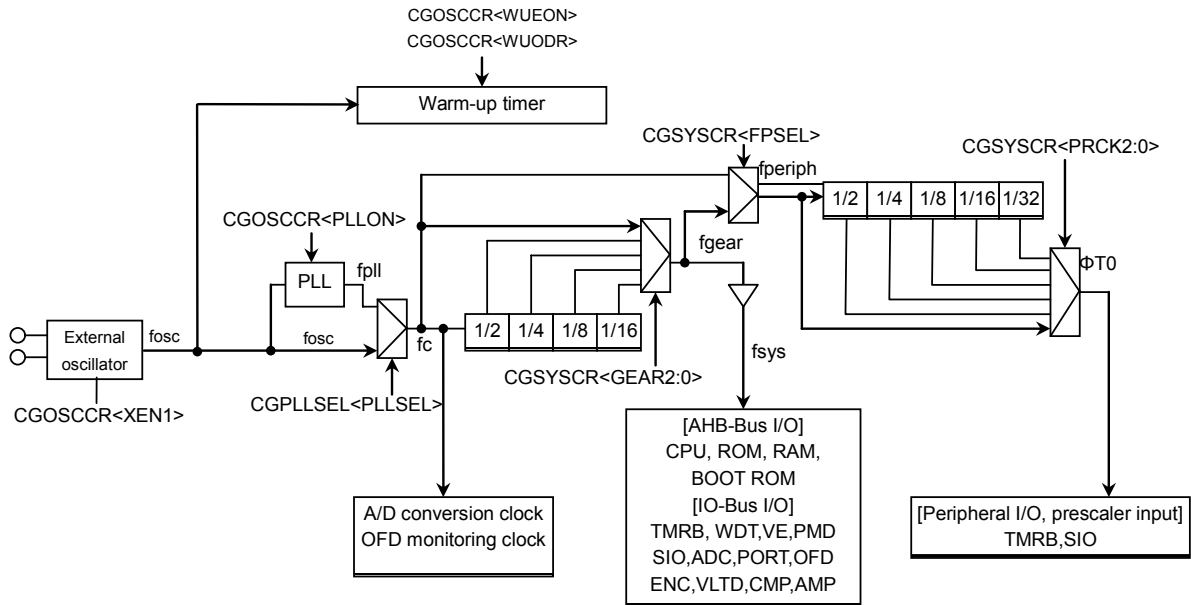


Fig.6-1 Clock Block Diagram

**(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  $f_{pll}$  clock that is octuple of the high-speed oscillator output clock,  $f_{osc}$ . 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. 200 $\mu$ s 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.

How to configure the warm-up function

Specify the count up clock for the warm-up counter in the CGOSCCR<WUPSEL1> bit. Write "0" to <WUPSEL1>.

The warm-up time can be specified by setting the CGOSCCR<WUODR13:0>. 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 = fosc)  
CGOSCCR<WUPSEL1>="0" : Write "0" to CGOSCCR<WUPSEL1>  
CGOSCCR<WUPODR13:0>= "Warm-up time" / (1/fosc) / 4  
: 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)** Always write "0" to bit15 and bit14 in CGOSCCR.

### 6.3.5 System Clock

The TMPM370 offers high-speed clock as system clock. The high-speed clock is dividable.

- Input frequency from X1 and X2: 8MHz to 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 |
| OSC             | 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 | **   |

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

Note 2 : Clock gear setting: available in CGSYSCR<GEAR2:0>

Note 3 : \*\*: Reserved

**(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.

### 6.3.6 Prescaler Clock Control

Each peripheral function (TMRB0-7 and SIO0-3) has a prescaler for dividing a clock. As the clock  $\phi T0$  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  $\phi T0$ .

**(Note)** To use the clock gear, ensure that you make the time setting such that prescaler output  $\phi Tn$  from each peripheral function is slower than fsys ( $\phi Tn < fsys$ ). 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”.

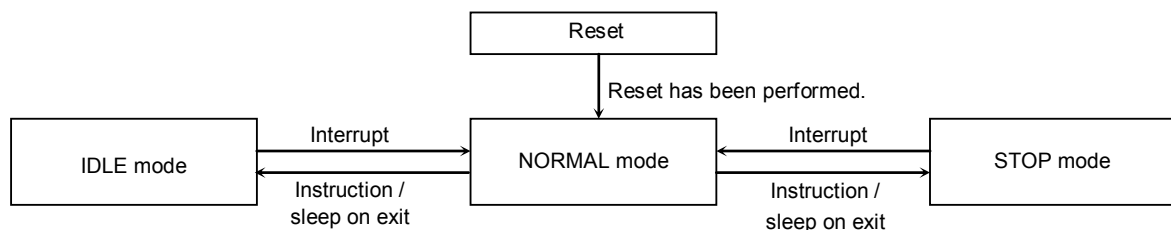


Fig.6-2 Mode Transition Diagram



## 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 TMPM370 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 TMPM370 does not offer any event for releasing the low power consumption mode.
- (Note 2)** The TMPM370 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.

**(Note)** 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

|           | Pin Name   | I/O         | <DRVE>=0  | <DRVE>=1  |
|-----------|--|-------------|---|---|
| Not Ports | X1   | Input only  | ×   | ×   |
|           | X2   | Output only | "H" level output  | "H" level output  |
|           | RESET, MODE  | Input only  | ○   | ○   |
| Ports     | PB3, PB5<br>[When used as a debug pin (PBFRC<n>=1) and output is enabled (PBCRC<n>=1)]   | Input       | ×   | Depends on PxIE<n>.   |
|           |  | Output      | Enabled when data is valid.<br>Disabled when data is invalid. | Enabled when data is valid.<br>Disabled when data is invalid. |
|           | PH0-2, PA0, PA2, PE4, PE6-7, PA7, PD3, PL0-1, PJ6-7, PK0-1<br>[When used as an interrupt pin (PxFR<n>=1) and input is enabled (PxIE<n>=1)] | Input       | ○   | ○   |
|           |  | Output      | ×   | Depends on PxCR<n>.   |
|           | Other port pins  | Input       | ×   | Depends on PxIE<n>.   |
|           |  | Output      | ×   | Depends on PxCR<n>.   |

○ : 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<br><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                        | ○      | ○                                       | ×          |
| ADC                        | ○      | ○                                       | ×          |
| VE                         | ○      | ON/OFF<br>selectable for<br>each module | ×          |
| SIO                        | ○      |   | ×          |
| TMRB                       | ○      |   | ×          |
| WDT                        | ○      |   | ×          |
| AMP/CMP                    | ○      | ○                                       | ○ (Note 2) |
| 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 | INT0 to INTF (Note 1)  | ○    | ○    |
|                            |           | INTRX0 to 3, INTTX0 to 3   | ○    | ×    |
|                            |           | INTVCNA, INTVCNB   | ○    | ×    |
|                            |           | INTEMG0 to 1   | ○    | ×    |
|                            |           | INTOVV0 to 1   | ○    | ×    |
|                            |           | INTTB0 to 7  | ○    | ×    |
|                            |           | INTPMD0 to 1   | ○    | ×    |
|                            |           | INTENC0 to 1   | ○    | ×    |
|                            |           | INTCAP00, 01, 10, 11, 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71 | ○    | ×    |
|                            |           | INTAD0PDA, INTAD1PDA, INTAD0PDB, INTAD1PDB                           | ○    | ×    |
|                            |           | INTAD0CPA, INTAD1CPA, INTAD0CPB, INTAD1CPB                           | ○    | ×    |
|                            |           | INTAD0SFT, INTAD1SFT   | ○    | ×    |
|                            |           | INTAD0TMR, INTAD1TMR   | ○    | ×    |
|                            |           | NMI (WDT)  | ○    | ×    |
| 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)** 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 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 may require the warm-up so that the internal oscillator provides stable oscillation.

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> and to set the warm-up time in the CGOSCCR<WUODR13:0> before executing the instruction to enter the STOP mode.

**(Note)** 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. 200 $\mu$ s 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.

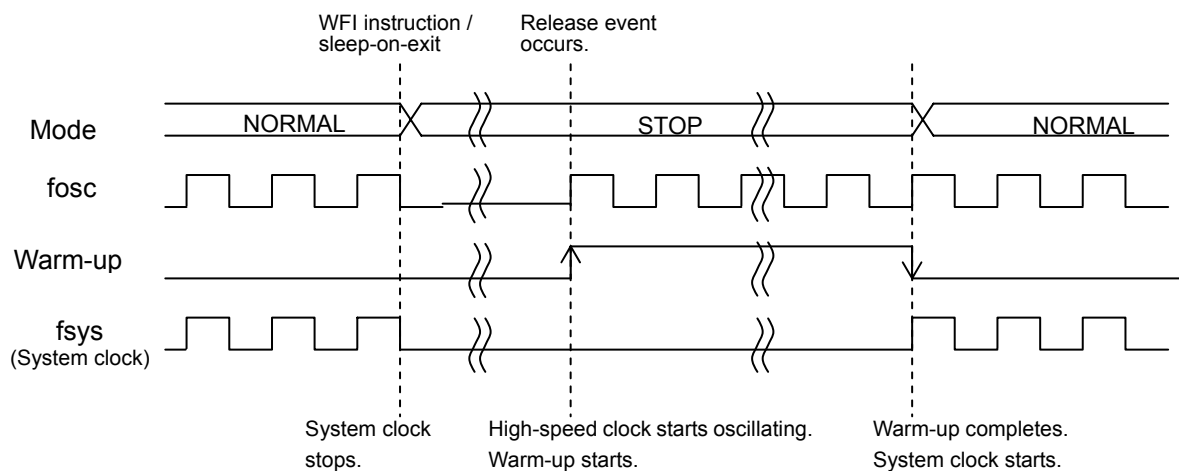
## 6.6.7 Clock Operations in Mode Transition

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

### 6.6.7.1 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. Exceptions

This chapter describes features, types and handling of exceptions.

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

### 7.1 Overview

An exception causes the CPU to stop the currently executing process and handle another process. There are two types of exceptions: those that are generated when some error condition occurs or when an instruction to generate an exception is executed; and those that are generated by hardware, such as an interrupt request signal from an external pin or peripheral function.

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

#### 7.1.1 Exception Types



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

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



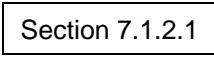
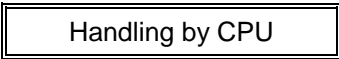

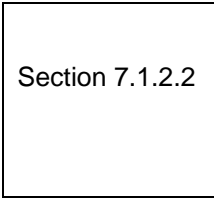
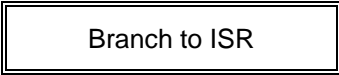

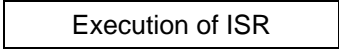

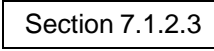
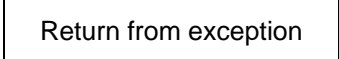
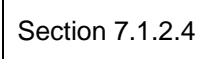
- Reset
- Non-Maskable Interrupt (NMI)
- Hard Fault
- Memory Management
- Bus Fault
- Usage Fault
- SVCcall (Supervisor Call)
- Debug Monitor
- PendSV
- SysTick
- External Interrupt

### 7.1.2 Handling Flowchart

The following shows how an exception/interrupt is handled.

 indicates hardware handling.  Indicates software handling.

Each step is described later in this chapter.

| Processing  | Description  | See   |
|---|--|---|
|  Detection by CG/CPU<br>  | The CG/CPU detects the exception request.                              |  Section 7.1.2.1   |
|  Handling by CPU<br>      | The CPU handles the exception request.                                 |  Section 7.1.2.2   |
|  Branch to ISR<br>        | The CPU branches to the corresponding interrupt service routine (ISR). |   |
|  Execution of ISR<br> | Necessary processing is executed.                                      |  Section 7.1.2.3 |
|  Return from exception   | The CPU branches to another ISR or returns to the previous program.    |  Section 7.1.2.4 |



### 7.1.2.1 Exception Request and Detection

#### (1) Exception occurrence

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

An exception by CPU instruction execution is caused when the CPU executes an instruction that generate 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 is generated from an external interrupt pin or peripheral function. For interrupts that are used for releasing a standby mode, relevant settings must be made in the clock generator. For details, refer to “7.5 Interrupts”.

#### (2) Exception detection

If multiple exceptions occur simultaneously, the CPU takes the exception with the highest priority. Table 7-1 shows the priority of exceptions. “Configurable” means that it is possible to assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 7-1 Exception Types and Priority

| No   | Exception type         | Priority     | Description  |
|------|------------------------|--------------|--|
| 1    | Reset                  | -3 (highest) | Reset pin, WDT ,POR, LVTD 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)<br>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   | SVCa l l               | 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   | SysT ick               | 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 “List of Interrupt Factors”.**

### (3) Priority setting

Use the Interrupt Priority Registers to assign a priority to each of the external interrupts. The priority of other exceptions can be set in the System Handler Priority Registers.

The priority registers are configurable, allowing the number of bits for setting priority levels to vary between three to eight bits. Therefore, the range of priority levels that can be assigned vary with each product.

In case of eight bits, a priority level can be set from 0 to 255. Priority level 0 is the highest priority level.

In case of the same priority level to multiple exceptions, the lowest-numbered exception has the highest priority.

**(Note) In this product, three bits are used for assigning a priority level in the Interrupt Priority Registers and System Handler Priority Registers.**

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

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

#### (1) Stacking

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

- Program Counter (PC)
- Program Status Register (xPSR)
- r0 - 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.

|               |            |
|---------------|------------|
| <b>Old SP</b> | <previous> |
|               | xPSR       |
|               | PC         |
|               | LR         |
|               | r12        |
|               | r3         |
|               | r2         |
|               | r1         |
| SP            | r0         |

#### (2) Fetching an ISR

At the same time as pushing the register contents to the stack, the CPU executes an instruction to fetch an ISR.

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.

It must always be set the first four words (stack top address, reset ISR address, NMI ISR address, and Hard Fault ISR address) .

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        | Bus Fault              | ISR address                     | Optional |
| 0x14        | Usage Fault            | ISR address                     | Optional |
| 0x18 ~ 0x24 | Reserved               |                                 |          |
| 0x28        | SVCa l l               | ISR address                     | Optional |
| 0x2C        | Debug Monitor          | ISR address                     | Optional |
| 0x30        | Reserved               |                                 |          |
| 0x34        | PendSV                 | ISR address                     | Optional |
| 0x38        | SysT i c k             | ISR address                     | Optional |
| 0x3C        | External Interrupt     | ISR address                     | Optional |

### 7.1.2.3 Executing an ISR

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

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

For details about interrupt handling, see “7.5 Interrupts”.

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

### 7.1.2.4 Exception exit

#### (1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

#### (2) Exception exit sequence

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

- Pop eight registers

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

- Load current active interrupt number

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

- Select SP

If returning to an exception (Handler Mode), SP is SP\_main. If returning to Thread Mode, SP can be SP\_main or SP\_process.

## 7.2 Reset Exceptions

Reset exceptions are generated from the following three 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 WDT

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

- Reset exception by SYSRESETREQ

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

## 7.3 Non-Maskable Interrupts (NMIs)

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

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

- Non-maskable interrupt by WDT

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

## 7.4 SysTick

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

In case 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. It is also possible to pend 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.

|   |
|---|
| <p><b>(Note)</b> In this product, the system timer counts based on a clock obtained by dividing the clock input from the X1 pin by 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.</p> |
|---|



## 7.5 Interrupts

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

The CPU is notified of interrupts by each signal of interrupt factor. It sets priority on the interrupts and handles an interrupt request with the highest priority.

The CPU is notified of the interrupt factor, which is used for clearing the standby modes, via a clock generator. Therefore setting of the clock generator is required.

### 7.5.1 Interrupt factors

#### 7.5.1.1 Interrupt route

Fig7.1 shows an interrupt request route.

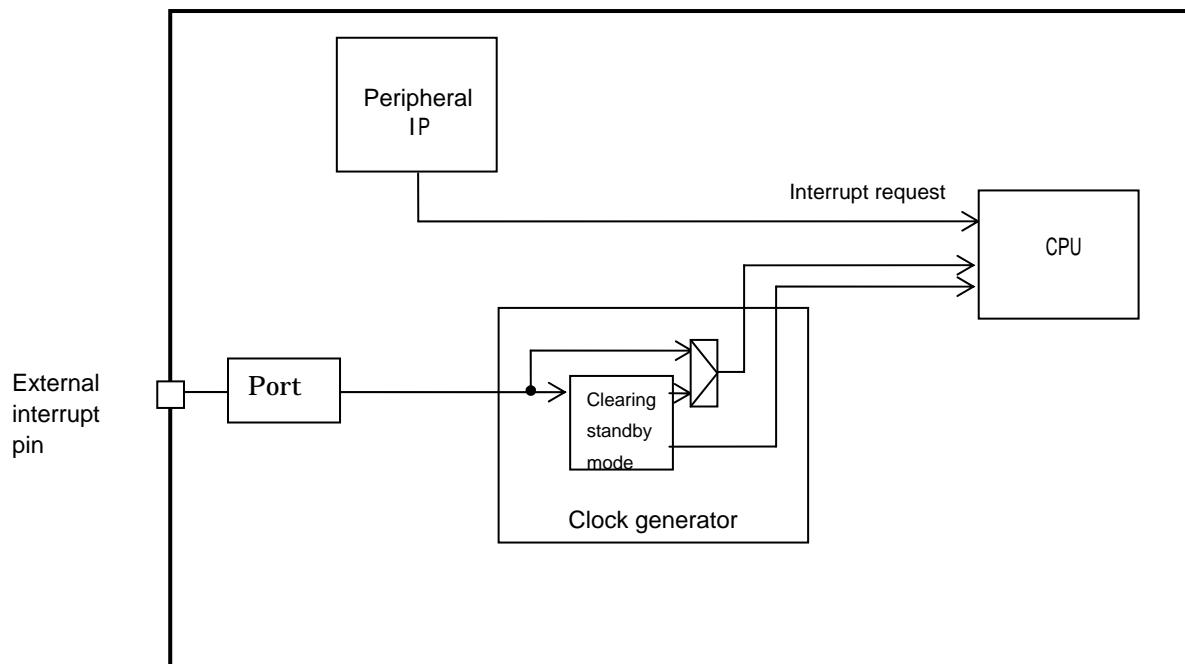


Fig 7.1 Interrupt Route

### 7.5.1.2 Interrupt signal generation

An interrupt request signal is generated from an external pin assigned as the interrupt factor or a peripheral IP.

- From external pin

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

- From peripheral IP

Set the peripheral IP to make it possible to output the interrupt.

See chapters of relevant IP for details.

### 7.5.1.3 Transmission of interrupt signal

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

An interrupt that can be used to clear the standby mode is transmitted to the CPU via the clock generator. Therefore, it is necessary to set the clock generator in advance. An interrupt from an external pin can be used without setting the clock generator if it does not function as the clearing standby factor.

## 7.5.1.4 List of Interrupt Factors

Table 7.2 shows the list of interrupt factors.

Table 7.2 List of Hardware Interrupt Factors (1/2)

| INT No | Interrupt factors |  | Active level<br>(Clearing standby) | CG interrupt mode control register |                                |         |
|--------|-------------------|--|------------------------------------|------------------------------------|--------------------------------|---------|
| 0      | INT0              | Interrupt Pin (PH0/AINA0/96pin or 98pin)         | High/Low Edge/Level Selectable     | CGIMCGA                            |                                |         |
| 1      | INT1              | Interrupt Pin (PH1/AINA1/95pin or 97pin)         |                                    |                                    |                                |         |
| 2      | INT2              | Interrupt Pin (PH2/AINA2/94pin or 96pin)         |                                    |                                    |                                |         |
| 3      | INT3              | Interrupt Pin (PA0/TB0IN/2pin or 4pin)           |                                    |                                    |                                |         |
| 4      | INT4              | Interrupt Pin (PA2/TB1IN/4pin or 6pin)           |                                    |                                    |                                |         |
| 5      | INT5              | Interrupt Pin (PE4/TB2IN//15pin or 17pin)        |                                    |                                    |                                |         |
| 6      | INTRX0            | Serial reception (channel.0)                     |                                    | CGIMCGB                            |                                |         |
| 7      | INTTX0            | Serial transmit (channel.0)                      |                                    |                                    |                                |         |
| 8      | INTRX1            | Serial reception (channel.1)                     |                                    |                                    |                                |         |
| 9      | INTTX1            | Serial transmit (channel.1)                      |                                    |                                    |                                |         |
| 10     | INTVCNA           | Vector Engine interrupt A                        |                                    |                                    |                                |         |
| 11     | INTVCNB           | Vector Engine interrupt B                        |                                    |                                    |                                |         |
| 12     | INTEMG0           | PMD0 EMG interrupt                               |                                    |                                    |                                |         |
| 13     | INTEMG1           | PMD1 EMG interrupt                               |                                    |                                    |                                |         |
| 14     | INTOVV0           | PMD0 OVV interrupt                               |                                    |                                    |                                |         |
| 15     | INTOVV1           | PMD1 OVV interrupt                               |                                    |                                    |                                |         |
| 16     | INTADAPDA         | ADCA conversion triggered by PMD0 is finished    |                                    |                                    |                                |         |
| 17     | INTADBPDA         | ADCB conversion triggered by PMD0 is finished    |                                    |                                    |                                |         |
| 18     | INTADAPDB         | ADCA conversion triggered by PMD1 is finished    |                                    |                                    |                                |         |
| 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     | INTPMD0           | PMD0 PWM interrupt                               |                                    |                                    |                                |         |
| 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     | INTCAP51          | 16bit TMRB5 input capture 1                      |                                    |                                    |                                |         |
| 38     | INT6              | Interrupt Pin (PE6/TB3IN//17pin or 19pin)        |                                    |                                    | High/Low Edge/Level Selectable | CGIMCGB |
| 39     | INT7              | Interrupt Pin (PE7/TB3OUT/18pin or 20pin)        |                                    |                                    |                                |         |
| 40     | INTRX2            | Serial reception (channel.2)                     |                                    |                                    |                                |         |
| 41     | INTTX2            | Serial transmit (channel.2)                      |                                    |                                    |                                |         |
| 42     | INTADACPA         | ADCA conversion monitoring function interrupt A  |                                    |                                    |                                |         |
| 43     | INTADBCPA         | ADCB conversion monitoring function interrupt A  |                                    |                                    |                                |         |
| 44     | INTADACPB         | ADCA conversion monitoring function interrupt B  |                                    |                                    |                                |         |
| 45     | INTADACPB         | ADCB conversion monitoring function interrupt B  |                                    |                                    |                                |         |

Table 7.2 List of Hardware Interrupt Factors (2/2)

| No. | Interrupt factors |   | Active trigger<br>(Clearing standby) | CG interrupt<br>mode control<br>register |                                      |         |
|-----|-------------------|---|--------------------------------------|--|--------------------------------------|---------|
| 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  | INTADASFT         | ADC unit A conversion started by software is finished |                                      |  |                                      |         |
| 55  | INTADBSFT         | ADC unit B conversion started by software is finished |                                      |  |                                      |         |
| 56  | INTADATMR         | ADC unit A conversion triggered by timer is finished  |                                      |  |                                      |         |
| 57  | INTADBTMR         | ADC unit B conversion triggered by timer is finished  |                                      |  |                                      |         |
| 58  | INT8              | Interrupt Pin (PA7/TB4IN/9pin or 11pin)               |                                      |  | High/Low<br>Edge/Level<br>Selectable | CGIMCGC |
| 59  | INT9              | Interrupt Pin (PD3/33pin or 35pin)                    |                                      |  |                                      |         |
| 60  | INTA              | Interrupt Pin (PL1/21pin or 23pin)                    |                                      |  |                                      |         |
| 61  | INTB              | Interrupt Pin (PL0/20pin or 22pin)                    |                                      |  |                                      |         |
| 62  | INTENC0           | Ender input0 interrupt                                |                                      |  |                                      |         |
| 63  | INTENC1           | Ender 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                           | High/Low<br>Edge/Level<br>Selectable | CGIMCGD                                  |                                      |         |
| 73  | INTCAP71          | 16bit TMRB7 input capture 1                           |                                      |  |                                      |         |
| 74  | INTC              | Interrupt Pin (PJ6/AINB9/74pin or 76 pin)             |                                      |  |                                      |         |
| 75  | INTD              | Interrupt Pin (PJ7/AINB10/73pin or 75pin)             |                                      |  |                                      |         |
| 76  | INTE              | Interrupt Pin (PK0/AINB11/72pin or 74pin)             |                                      |  |                                      |         |
| 77  | INTF              | Interrupt Pin (PK1/AINB12/71pin or 73pin)             |                                      |  |                                      |         |

#### 7.5.1.5 Active Level

The active level indicates which change in signal of an interrupt factor triggers an interrupt. The CPU recognize an interrupt signal as an interrupt factor when it is changed from “L” to “H”. A signal directly sent from the peripheral IP to the CPU is configured to output the “H” pulse as an interrupt request.

Only interrupt request from external pin have the option as the interrupt to clear the standby mode. The active level setting for Clock Generator is selectable from “H” level, “L” level, rising edge or falling edge.

If the interrupt is used for clearing the standby mode, setting the clock generator register is required.  
i.e. Enable the CGIMCGx<INTxEN> bit and specify the active level in the CGIMCGx<EMCG2:0> bits.



The interrupt detected by the clock generator is notified to the CPU as “H” level signal.

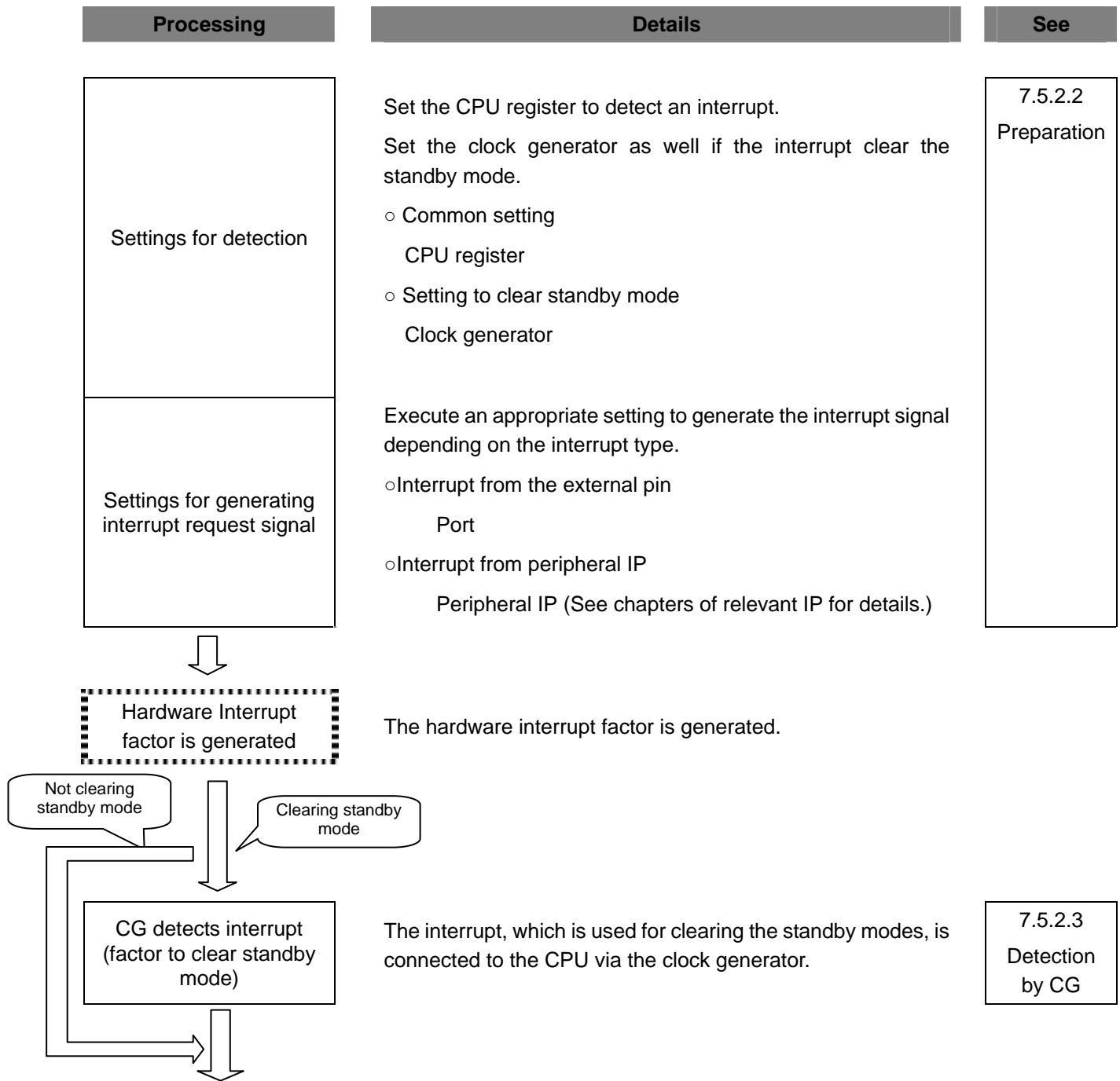
An interrupt from the external pin can be used without setting the clock generator in case it does not function as the standby clearing factor. However, inputting the “H” pulse or the “H” level signal is required so that the CPU can detect it as an interrupt factor.

## 7.5.2 Interrupt handling

### 7.5.2.1 Flowchart

The following shows how an interrupt is handled.

 indicates hardware handling.  indicates software handling.



| Processing  | Details   | See   |
|---|---|---|
| <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 10px; text-align: center;">Detecting interrupt</div>                     | <p>The CPU detects the interrupt.</p> <p>In case several interrupt factor are detected, the interrupt factor with the highest priority is selected according to the priority order.</p> | <div style="border: 1px solid black; padding: 5px;">7.5.2.4<br/>Detection<br/>by CPU</div>              |
| <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 10px; text-align: center;">Handling interrupt</div>                      | <p>The CPU handles the interrupt.</p> <p>The CPU pushes register contents to the stack before entering the interrupt service routine</p>  | <div style="border: 1px solid black; padding: 5px;">7.5.2.5<br/>CPU<br/>processing</div>                |
| <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 10px; text-align: center;">Executing interrupt<br/>service routine</div> | <p>Program for the interrupt service routine.<br/>Clear the interrupt factor if needed.</p>   | <div style="border: 1px solid black; padding: 5px;">7.5.2.6<br/>Interrupt<br/>service<br/>routine</div> |
| <div style="text-align: center;">↓</div> <div style="border: 1px solid black; padding: 10px; text-align: center;">Returning to preceding<br/>program</div>      | <p>Configure to return to the preceding program from the interrupt service routine.</p>   |   |

### 7.5.2.2 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.

In order not to generate unnecessary interrupt after condition setting, In case of setting the clock generator, it need to clear the interrupt related data in the clock generator before enable the interrupt.

The following shows 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 IP )
- (5) Configuring the clock generator
- (6) 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 Interrupt Clear-Enable register. Each bit of the register, of which default setting is "0" with interrupt disabled, is assigned to single interrupt factor.

|                           |   |                        |
|---------------------------|---|------------------------|
| ●CPU register             |   |                        |
| Interrupt Clear-Enable<m> | ← | "1"(disable interrupt) |

(Note) m: corresponding bit.

#### ( 2 ) CPU interrupt registers setting

Assign a interrupt priority level from 0 to 255 by writing to the eight bit field in an Interrupt Priority Register.

Priority level 0 is the highest priority level.

|                       |   |            |
|-----------------------|---|------------|
| ●CPU register         |   |            |
| Interrupt Priority<m> | ← | "Priority" |

(Note) m: corresponding bit.

## ( 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<br>m: corresponding bit<br>x: function register number |
|--------|---|

## ( 4 ) Preconfiguration 2 (interrupt from peripheral IP)

The setting varies depending on the IP to be used. See chapters of relevant IP for details.

## ( 5 ) Configuring the clock generator

For an interrupt to clear the standby mode, configure active level and enabling interrupt by the CGIMCG register of the clock generator. The CGIMCG registers are the registers for configuring each factor.

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt. To clear corresponding interrupt request, write value corresponding to the interrupt to be used to the CGICRCG register. See 7.6.3.5 CGICRCG Interrupt Clear Register for each value.

An interrupt from the external pin can be used without setting the clock generator if it does not function as the standby clearing factor. However, inputting the "H" pulse or the signal in "H" level is required so that the CPU can detect it as an interrupt factor.

|                           |   |   |
|---------------------------|---|---|
| ●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<br>m: number assigned to each interrupt factor |
|--------|---|



## ( 6 ) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

It is possible to clear the suspended interrupt by writing the Interrupt Clear-Pending register. Then, enable the intended interrupt with the Interrupt Set-Enable register. Each bit of the register is assigned to each interrupt factor.

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

|                            |   |     |
|----------------------------|---|-----|
| ●CPU register              |   |     |
| Interrupt Clear-Pending<m> | ← | "1" |
| Interrupt Set-Enable<m>    | ← | "1" |

|                             |
|-----------------------------|
| (Note) m: corresponding bit |
|-----------------------------|

### 7.5.2.3 Detection by Clock Generator

If the interrupt is used for clearing the standby mode, the interrupt factor is detected by an active level specified in the clock generator, and notified to the CPU.

The interrupt active level triggered by a rising or falling edge is kept in the clock generator after detection. However, if "H" or "L" level signal is specified as the trigger to enter the active state, the CPU considers that the interrupt factor is cleared upon exiting from the active state. Therefore, the active state needs to be kept until the interrupt is detected.

The clock generator notified to the CPU the interrupt detected by "H" level signal. The CPU considers the interrupt signal as an interrupt factor when it is changed from "L" to "H". To generate an interrupt again, the factor held in the clock generator needs to be cleared with the CGICRCG clear register.

### 7.5.2.4 Detection by CPU

The CPU detects an interrupt factor with the highest priority.

### 7.5.2.5 CPU processing

On detecting the interrupt, the CPU pushes the contents of PC, PSR, r0-r3, r12 and LR to the stack before entering the interrupt service routine.

### 7.5.2.6 Interrupt Service Routine

Interrupt service routine 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 interrupt service routine

Common interrupt service routine is accompanied with the interrupt handling and the pushing of the register contents. 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.

An interrupt with the higher priority and faults such as NMI are accepted even when the interrupt service routine is being executed. Therefore, it is recommend to push the contents of the general purpose register that might be rewritten.

#### ( 2 ) Clearing interrupt factor

As for an interrupt factor clearing the standby mode, it need to clear the interrupt request with the CGICRCG clear register of the clock generator.

If “H” or “L” level signal is specified as the trigger to enter the active state, the factor is held unless it is cleared. In this case, the factor needs to be cleared. Clearing the factor causes clearing the interrupt request signal from clock generator.

If a rising or falling edge is specified as the trigger to enter the active state, the factor is cleared by setting the value which corresponds to the interrupt, to the CGICRCG clear register. The factor is detected again when the specified edge appears again.

## 7.6 Exception/Interrupt-related registers

The clock generator registers and their addresses are as shown below.

### 7.6.1 Register List

| NVIC Registers                            |                                     |
|---|-------------------------------------|
| SysTick Control and Status Register       | 0xE000_E010                         |
| SysTick Reload Value Register             | 0xE000_E014                         |
| SysTick Current Value Register            | 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                         |
| 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 |

7.6.2 NVIC Registers

7.6.2.1 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<br>1: Core clock | 0: Do not pend SysTick<br>1: Pend SysTick | 0: Disable<br>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<br>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.

7.6.2.2 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 input from the X1 pin by 32.

7.6.2.3 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<br>[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<br>[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<br>[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.

7.6.2.4 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<br>1: No reference clock | 0: Calibration value is 10 ms.<br>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 input from the X1 pin by 32.  
 The SysTick Calibration Value Register is set to a value that 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.

## 7.6.2.5 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<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 6<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 5<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 4<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 3<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 2<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 1<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 0<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  |
|             | 15   | 14   | 13   | 12   | 11   | 10   | 9  | 8  |
| bit Symbol  | SETENA   |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |
| After reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Function    | Interrupt number 15<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 14<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 13<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 12<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 11<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 10<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 9<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 8<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>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<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 22<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 21<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 20<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 19<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 18<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 17<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 16<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled |
|             | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   |
| bit Symbol  | SETENA   |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |
| After reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Function    | Interrupt number 31<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 30<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 29<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 28<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 27<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 26<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 25<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 24<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>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 7.5.1.4 List of Interrupt Factors.**



7.6.2.6 Interrupt Set-Enable Register 2

|             |  |  |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|--|--|
|             | 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 39<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 38<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 37<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 36<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 35<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 34<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 33<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 32<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled |  |
|             | 15   | 14   | 13   | 12   | 11   | 10   | 9  | 8  |  |
| bit Symbol  | SETENA   |  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |  |
| After reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |
| Function    | Interrupt number 47<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 46<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 45<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 44<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 43<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 42<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 41<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 40<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>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 55<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 54<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 53<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 52<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 51<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 50<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 49<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 48<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled |  |
|             | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   |  |
| bit Symbol  | SETENA   |  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |  |
| After reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |
| Function    | Interrupt number 63<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 62<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 61<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 60<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 59<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 58<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 57<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 56<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>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 7.5.1.4 List of Interrupt Factors.

7.6.2.7 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<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 70<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 69<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 68<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 67<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 66<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 65<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 64<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>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<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 76<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 75<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 74<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 73<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 72<br>[Write]<br>1: Enable<br>[Read]<br>0: Disabled<br>1: Enabled |  |
|             | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |  |
| bit Symbol  | "0" is read.   |  |  |  |  |  |  |  |  |
| 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  | "0" is read.   |  |  |  |  |  |  |  |  |
| 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 Section 7.5.1.4 List of Interrupt Factors.

7.6.2.8 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<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 6<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 5<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 4<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 3<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 2<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 1<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 0<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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    | Interrupt number 15<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 14<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 13<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 12<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 11<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 10<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 9<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled  | Interrupt number 8<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 22<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 21<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 20<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 19<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 18<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 17<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 16<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled |
|             | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| bit Symbol  | CLRENA  |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Function    | Interrupt number 31<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 30<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 29<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 28<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 27<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 26<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 25<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 24<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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 Section 7.5.1.4 List of Interrupt Factors.**

7.6.2.9 Interrupt Clear-Enable Register 2

|             |   |   |   |   |   |   |   |   |
|-------------|---|---|---|---|---|---|---|---|
|             | 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 39<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 38<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 37<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 36<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 35<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 34<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 33<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 32<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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    | Interrupt number 47<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 46<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 45<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 44<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 43<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 42<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 41<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 40<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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 55<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 54<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 53<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 52<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 51<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 50<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 49<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 48<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled |
|             | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| bit Symbol  | CLRENA  |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Function    | Interrupt number 63<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 62<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 61<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 60<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 59<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 58<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 57<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 56<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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 Section 7.5.1.4 List of Interrupt Factors.**

7.6.2.10 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<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 70<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 69<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 68<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 67<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 66<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 65<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 64<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 76<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 75<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 74<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 73<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>1: Enabled | Interrupt number 72<br>[Write]<br>1: Disable<br>[Read]<br>0: Disabled<br>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 Section 7.5.1.4 List of Interrupt Factors.**

7.6.2.11 Interrupt Set-Pending Register 1

|             |   |   |   |   |   |   |   |   |
|-------------|---|---|---|---|---|---|---|---|
|             | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 7<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 6<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 5<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 4<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 3<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 2<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 1<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 0<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  |
|             | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 15<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 14<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 13<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 12<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 11<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 10<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 9<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 8<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending  |
|             | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 23<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 22<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 21<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 20<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 19<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 18<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 17<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 16<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 31<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 30<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 29<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 28<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 27<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 26<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 25<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 24<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>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.

|   |
|---|
| <b>C</b> For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors. |
|---|

7.6.2.12 Interrupt Set-Pending Register 2

|             |   |   |   |   |   |   |   |   |
|-------------|---|---|---|---|---|---|---|---|
|             | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 39<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 38<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 37<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 36<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 35<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 34<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 33<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 32<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 47<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 46<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 45<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 44<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 43<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 42<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 41<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 40<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 55<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 54<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 53<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 52<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 51<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 50<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 49<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 48<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| bit Symbol  | SETPEND   |   |   |   |   |   |   |   |
| Read/Write  | R/W   |   |   |   |   |   |   |   |
| After reset | Undefined   |   |   |   |   |   |   |   |
| Function    | Interrupt number 63<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 62<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 61<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 60<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 59<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 58<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 57<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 56<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>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.

|   |
|---|
| <b>C</b> For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors. |
|---|

7.6.2.13 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<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 70<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 69<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 68<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 67<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 66<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 65<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 64<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>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<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 76<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 75<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 74<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 73<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 72<br>[Write]<br>1: Pend<br>[Read]<br>0: Not pending<br>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 Section 7.5.1.4 List of Interrupt Factors.**

7.6.2.14 Interrupt Clear-Pending Register 1

|             |  |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|--|
|             | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |
| Function    | Interrupt number 7<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 6<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 5<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 4<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 3<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 2<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 1<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 0<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  |
|             | 15   | 14   | 13   | 12   | 11   | 10   | 9  | 8  |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |
| Function    | Interrupt number 15<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 14<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 13<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 12<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 11<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 10<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 9<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  | Interrupt number 8<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending  |
|             | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |
| Function    | Interrupt number 23<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 22<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 21<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 20<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 19<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 18<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 17<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 16<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |
| Function    | Interrupt number 31<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 30<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 29<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 28<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 27<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 26<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 25<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 24<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>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 Section 7.5.1.4 List of Interrupt Factors.**

7.6.2.15 Interrupt Clear-Pending Register 2

|             |  |  |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|--|--|
|             | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |  |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |  |
| Function    | Interrupt number 39<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 38<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 37<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 36<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 35<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 34<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 33<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 32<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 32<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 15   | 14   | 13   | 12   | 11   | 10   | 9  | 8  |  |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |  |
| Function    | Interrupt number 47<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 46<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 45<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 44<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 43<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 42<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 41<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 40<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 40<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |  |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |  |
| Function    | Interrupt number 55<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 54<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 53<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 52<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 51<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 50<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 49<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 48<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 48<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending |
|             | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   |  |
| bit Symbol  | CLRPEND  |  |  |  |  |  |  |  |  |
| Read/Write  | R/W  |  |  |  |  |  |  |  |  |
| After reset | Undefined  |  |  |  |  |  |  |  |  |
| Function    | Interrupt number 63<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 62<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 61<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 60<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 59<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 58<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 57<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 56<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 56<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>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

|  |
|--|
| <b>(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors.</b> |
|--|

7.6.2.16 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<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 38<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 37<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 36<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 35<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 34<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 33<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 32<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>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<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 44<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 43<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 42<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 41<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>1: Pending | Interrupt number 40<br>[Write]<br>1: Clear pending interrupt<br>[Read]<br>0: Not pending<br>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 Section 7.5.1.4 List of Interrupt Factors.

### 7.6.2.17 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  |    | PRI_2  |    | PRI_1  |   | PRI_0  |
| 0xE000_E404 |    | PRI_7  |    | PRI_6  |    | PRI_5  |   | PRI_4  |
| 0xE000_E408 |    | PRI_11 |    | PRI_10 |    | PRI_9  |   | PRI_8  |
| 0xE000_E40C |    | PRI_15 |    | PRI_14 |    | PRI_13 |   | PRI_12 |
| 0xE000_E410 |    | PRI_19 |    | PRI_18 |    | PRI_17 |   | PRI_16 |
| 0xE000_E414 |    | PRI_23 |    | PRI_22 |    | PRI_21 |   | PRI_20 |
| 0xE000_E418 |    | PRI_27 |    | PRI_26 |    | PRI_25 |   | PRI_24 |
| 0xE000_E41C |    | PRI_31 |    | PRI_30 |    | PRI_29 |   | PRI_28 |
| 0xE000_E420 |    | PRI_35 |    | PRI_34 |    | PRI_33 |   | PRI_32 |
| 0xE000_E424 |    | PRI_39 |    | PRI_38 |    | PRI_37 |   | PRI_36 |
| 0xE000_E428 |    | PRI_43 |    | PRI_42 |    | PRI_41 |   | PRI_40 |
| 0xE000_E42C |    | PRI_47 |    | PRI_46 |    | PRI_45 |   | PRI_44 |
| 0xE000_E430 |    | PRI_51 |    | PRI_50 |    | PRI_49 |   | PRI_48 |
| 0xE000_E434 |    | PRI_55 |    | PRI_54 |    | PRI_53 |   | PRI_52 |
| 0xE000_E438 |    | PRI_59 |    | PRI_58 |    | PRI_57 |   | PRI_56 |
| 0xE000_E43C |    | PRI_63 |    | PRI_62 |    | PRI_61 |   | PRI_60 |
| 0xE000_E440 |    | PRI_67 |    | PRI_66 |    | PRI_65 |   | PRI_64 |
| 0xE000_E444 |    | PRI_71 |    | PRI_70 |    | PRI_69 |   | PRI_68 |
| 0xE000_E448 |    | PRI_75 |    | PRI_74 |    | PRI_73 |   | PRI_72 |
| 0xE000_E44C |    | -      |    | -      |    | PRI_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 0 to 3. The Interrupt Priority Registers for all other interrupt numbers have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.



|             | 7                              | 6  | 5  | 4            | 3  | 2  | 1  | 0  |
|-------------|--------------------------------|----|----|--------------|----|----|----|----|
| bit Symbol  | PRI_0                          |    |    |              |    |    |    |    |
| Read/Write  | R/W                            |    |    | R            |    |    |    |    |
| After reset | 0                              |    |    | 0            |    |    |    |    |
| Function    | Priority of interrupt number 0 |    |    | "0" is read. |    |    |    |    |
|             | 15                             | 14 | 13 | 12           | 11 | 10 | 9  | 8  |
| bit Symbol  | PRI_1                          |    |    |              |    |    |    |    |
| Read/Write  | R/W                            |    |    | R            |    |    |    |    |
| After reset | 0                              |    |    | 0            |    |    |    |    |
| Function    | Priority of interrupt number 1 |    |    | "0" is read. |    |    |    |    |
|             | 23                             | 22 | 21 | 20           | 19 | 18 | 17 | 16 |
| bit Symbol  | PRI_2                          |    |    |              |    |    |    |    |
| Read/Write  | R/W                            |    |    | R            |    |    |    |    |
| After reset | 0                              |    |    | 0            |    |    |    |    |
| Function    | Priority of interrupt number 2 |    |    | "0" is read. |    |    |    |    |
|             | 31                             | 30 | 29 | 28           | 27 | 26 | 25 | 24 |
| bit Symbol  | PRI_3                          |    |    |              |    |    |    |    |
| Read/Write  | R/W                            |    |    | R            |    |    |    |    |
| After reset | 0                              |    |    | 0            |    |    |    |    |
| Function    | Priority of interrupt number 3 |    |    | "0" is read. |    |    |    |    |

- <bit7:5> <PRI\_0> Priority of interrupt number 0
- <bit15:13> <PRI\_1> Priority of interrupt number 1
- <bit23:21> <PRI\_2> Priority of interrupt number 2
- <bit31:29> <PRI\_3> Priority of interrupt number 3

7.6.2.18 Vector Table Offset Register

|             |                           |    |            |              |    |    |    |    |  |
|-------------|---------------------------|----|------------|--------------|----|----|----|----|--|
|             | 7                         | 6  | 5          | 4            | 3  | 2  | 1  | 0  |  |
| bit Symbol  | TBLOFF                    |    |            |              |    |    |    |    |  |
| Read/Write  | R                         |    |            |              |    |    |    |    |  |
| After reset | 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

**7.6.2.19 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<br>(Usage Fault) |    | PRI_5<br>(Bus Fault) |   | PRI_4<br>(Memory Management) |   |
| 0xE000_ED1C | PRI_11<br>(SVCall)  |    | PRI_10                 |    | PRI_9                |   | PRI_8                        |   |
| 0xE000_ED20 | PRI_15<br>(SysTick) |    | PRI_14<br>(PendSV)     |    | PRI_13               |   | PRI_12<br>(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. |    |    |    |    |

7.6.2.20 System Handler Control and State Register

|             |   |  |   |   |   |   |                                       |  |
|-------------|---|--|---|---|---|---|---------------------------------------|--|
|             | 7   | 6  | 5   | 4   | 3                                       | 2   | 1                                     | 0  |
| bit Symbol  | SVCALL<br>ACT                             |  |   |   | USGFAU<br>LT<br>ACT                     |   | BUSFAU<br>LT<br>ACT                   | MEMFAU<br>LT<br>ACT                              |
| Read/Write  | R/W                                       | R  |   |   | R/W                                     | R   | R/W                                   | R/W  |
| After reset | 0   | 0  |   |   | 0                                       | 0   | 0                                     | 0  |
| Function    | SVCall<br>0: Inactive<br>1: Active        | "0" is read.                                 |   |   | Usage fault<br>0: Inactive<br>1: Active | "0" is read.                              | Bus fault<br>0: Inactive<br>1: Active | Memory<br>Management<br>0: Inactive<br>1: Active |
|             | 15  | 14   | 13  | 12  | 11                                      | 10  | 9                                     | 8  |
| bit Symbol  | SVCALL<br>PENDEDED                        | BUSFAU<br>LT<br>PENDEDED                     | MEMFAU<br>LT<br>PENDEDED                                | USGFAU<br>LT<br>PENDEDED                          | SYSTICK<br>ACT                          | PENDSV<br>ACT                             |                                       | MONITO<br>R<br>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<br>0: Not<br>pending<br>1: Pending | Bus Fault<br>0: Not<br>pending<br>1: Pending | Memory<br>Management<br>0: Not<br>pending<br>1: Pending | Usage<br>Fault<br>0: Not<br>pending<br>1: Pending | SysTick<br>0: Inactive<br>1: Active     | PendSV<br>0: Inactive<br>1: Active        | "0" is read.                          | Debug<br>Monitor<br>0: Inactive<br>1: Active     |
|             | 23  | 22   | 21  | 20  | 19                                      | 18  | 17                                    | 16   |
| bit Symbol  |   |  |   |   |   | USGFAU<br>LT<br>ENA                       | BUSFAU<br>LT<br>ENA                   | MEMFAU<br>LT<br>ENA                              |
| Read/Write  | R   |  |   |   |   | R/W                                       | R/W                                   | R/W  |
| After reset | 0   |  |   |   |   | 0   | 0                                     | 0  |
| Function    | "0" is read.                              |  |   |   |   | Usage<br>Fault<br>0: Disable<br>1: Enable | Bus Fault<br>0: Disable<br>1: Enable  | Memory<br>Management<br>0: Disable<br>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> <USGFAULTPENDEDED> Reads as "1" if Usage Fault is pending.
- <bit13> <MEMFAULTPENDEDED> Reads as "1" if Memory Management is pending.
- <bit14> <BUSFAULTPENDEDED> Reads as "1" if Bus Fault is pending.
- <bit15> <SVCALLPENDEDED> Reads as "1" if SVCAll is pending.

|         |               |   |
|---------|---------------|---|
| <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.**

7.6.3 Clock Generator Registers

7.6.3.1 CG Interrupt Mode Control Register A

This register set the clearing standby request active level of external interrupt INT0~INT3.

|             |             |   |   |        |  |  |  |   |   |
|-------------|-------------|---|---|--------|--|--|--|---|---|
| CGIMCGA     |             | 7   | 6   | 5      | 4  | 3  | 2  | 1   | 0   |
|             | bit Symbol  |   | EMCG02  | EMCG01 | EMCG00   | EMST01   | EMST00                                   | INT0F                                       | INT0EN                                      |
|             | 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 INT0 standby clear request. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |        |  | Active state of INT0 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened    | INT0 clear input<br>0: Disable<br>1: Enable |
|             |             | 15  | 14  | 13     | 12   | 11   | 10                                       | 9   | 8   |
| bit Symbol  |             | EMCG12  | EMCG11  | EMCG10 | EMST11   | EMST10   | INT1F                                    | INT1EN                                      |   |
| 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 INT1 standby clear request. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT1 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT1 clear input<br>0: Disable<br>1: Enable |   |
|             |             | 23  | 22  | 21     | 20   | 19   | 18                                       | 17  | 16  |
| bit Symbol  |             | EMCG22  | EMCG21  | EMCG20 | EMST21   | EMST20   | INT2F                                    | INT2EN                                      |   |
| 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 INT2 standby clear request. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT2 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT2 clear input<br>0: Disable<br>1: Enable |   |
|             |             | 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. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT3 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT3 clear input<br>0: Disable<br>1: Enable |   |

- (Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.
- (Note2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

7.6.3.2 CG Interrupt Mode Control Register B

This register set the clearing standby request active level of external interrupt INT4~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. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |        |  | Active state of INT4 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened    | INT4 clear input<br>0: Disable<br>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. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT5 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT5 clear input<br>0: Disable<br>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. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT6 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT6 clear input<br>0: Disable<br>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. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT7 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT7 clear input<br>0: Disable<br>1: Enable |   |

- (Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.
- (Note2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

7.6.3.3 CG Interrupt Mode Control Register C

This register set the clearing standby request active level of external interrupt INT8~INTB.

|             |             |   |   |        |  |  |  |   |   |
|-------------|-------------|---|---|--------|--|--|--|---|---|
| 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. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |        |  | Active state of INT8 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened    | INT8 clear input<br>0: Disable<br>1: Enable |
|             |             | 15  | 14  | 13     | 12   | 11   | 10                                       | 9   | 8   |
| bit Symbol  |             | EMCG92  | EMCG91  | EMCG90 | EMST91   | EMST90   | INT9F                                    | INT9EN                                      |   |
| 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 INT9 standby clear request. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INT9 standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INT9 clear input<br>0: Disable<br>1: Enable |   |
|             |             | 23  | 22  | 21     | 20   | 19   | 18                                       | 17  | 16  |
| bit Symbol  |             | EMCGA2  | EMCGA1  | EMCGA0 | EMSTA1   | EMSTA0   | INTAF                                    | INTAEN                                      |   |
| 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 INTA standby clear request. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INTA standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INTA clear input<br>0: Disable<br>1: Enable |   |
|             |             | 31  | 30  | 29     | 28   | 27   | 26                                       | 25  | 24  |
| bit Symbol  |             | EMCGB2  | EMCGB1  | EMCGB0 | EMSTB1   | EMSTB0   | INTBF                                    | INTBEN                                      |   |
| 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 INTB standby clear request. (101~111: setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |   |        | Active state of INTB standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |  | Interrupt happened<br>0:No<br>1:happened | INTB clear input<br>0: Disable<br>1: Enable |   |

- (Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.
- (Note2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.



7.6.3.4 CG Interrupt Mode Control Register D

This register set the clearing standby request active level of external interrupt INTC~INTF.

|         |             |             |   |    |    |  |    |    |  |   |
|---------|-------------|-------------|---|----|----|--|----|----|--|---|
| CGIMCGD | bit Symbol  | 7           | 6   | 5  | 4  | 3  | 2  | 1  | 0  |   |
|         | 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. (101~111; setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |    |    | Active state of INTC standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |    |    | Interrupt happened<br>0:No<br>1:happened | INTC clear input<br>0: Disable<br>1: Enable |
|         |             |             |   |    |    |  |    |    |  |   |
|         | bit Symbol  | 15          | 14  | 13 | 12 | 11   | 10 | 9  | 8  |   |
|         | 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. (101~111; setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |    |    | Active state of INTD standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |    |    | Interrupt happened<br>0:No<br>1:happened | INTD clear input<br>0: Disable<br>1: Enable |
|         |             |             |   |    |    |  |    |    |  |   |
|         | bit Symbol  | 23          | 22  | 21 | 20 | 19   | 18 | 17 | 16                                       |   |
|         | 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. (101~111; setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |    |    | Active state of INTE standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |    |    | Interrupt happened<br>0:No<br>1:happened | INTE clear input<br>0: Disable<br>1: Enable |
|         |             |             |   |    |    |  |    |    |  |   |
|         | bit Symbol  | 31          | 30  | 29 | 28 | 27   | 26 | 25 | 24                                       |   |
|         | 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. (101~111; setting prohibited)<br>000: "L" level<br>001: "H" level<br>010: Falling edge<br>011: Rising edge<br>100: Both edges |    |    | Active state of INTF standby clear request<br>00: -<br>01: Rising edge<br>10: Falling edge<br>11: Both edges |    |    | Interrupt happened<br>0:No<br>1:happened | INTF clear input<br>0: Disable<br>1: Enable |
|         |             |             |   |    |    |  |    |    |  |   |

- (Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.
- (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 interrupt 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 , 16 interrupt factors (INT0 to INTF) are available. CGIMCGA~CGIMCGD Registers 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, INT0 to INTF can be used as a normal interrupt without setting CGIMCGA~CGIMCGD Registers in CG.

7.6.3.5 CG Interrupt Clear Request Register

This register clear the Interrupt request from INT0~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<br>0_0000:INT0                      0_1000:INT8<br>0_0001:INT1                      0_1001:INT9<br>0_0010:INT2                      0_1010:INTA<br>0_0011:INT3                      0_1011:INTB<br>0_0100:INT4                      0_1100:INTC<br>0_0101:INT5                      0_1101:INTD<br>0_0110:INT6                      0_1110:INTE<br>0_0111:INT7                      0_1111:INTF<br>1_0000 ~ 1_1111: setting prohibited<br>* "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.  |    |    |   |        |        |        |        |

7.6.3.6 NMI Flag Register

NMI Flag register is a register for reading NMI generation status.

|             |              |              |    |    |    |    |    |    |         |
|-------------|--------------|--------------|----|----|----|----|----|----|---------|
| CGNMIFLG    |              | 7            | 6  | 5  | 4  | 3  | 2  | 1  | 0       |
|             | bit Symbol   |              |    |    |    |    | -  | -  | NMIFLG0 |
|             | Read/Write   | R            |    |    |    |    |    |    |         |
|             | After reset  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0       |
|             | Function     | "0" is read. |    |    |    |    |    | -  | -       |
|             |              | 15           | 14 | 13 | 12 | 11 | 10 | 9  | 8       |
| bit Symbol  |              |              |    |    |    |    |    |    |         |
| Read/Write  | R            |              |    |    |    |    |    |    |         |
| After reset | 0            | 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  | 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  | 0       |
| Function    | "0" is read. |              |    |    |    |    |    |    |         |

(Note) <NMIFLG0> is cleared to "0" when it is read.

7.6.3.7 Reset Flag Register

Reset Flag Register is a register for reading internal Reset generation status per generation factors. Since this register is not cleared automatically, it is necessary to write “0” to clear the register.

|             |              |    |  |   |   |  |   |  |         |
|-------------|--------------|----|--|---|---|--|---|--|---------|
| CGRSTFLG    |              | 7  | 6  | 5   | 4   | 3  | 2   | 1  | 0       |
|             | bit Symbol   |    |  | OFDRSTF   | DBGIRSTF  | VLTDIRSTF  | WDRSTF  | PINRSTF  | PONRSTF |
|             | Read/Write   | R  |  | R/W   | R/W   | R/W  | R/W   | R/W  | R/W     |
|             | After reset  | 0  | 0  | 0   | 0   | 0  | 0   | 0  | 1       |
| Function    | “0” is read. |    | OFD reset flag<br>0 : “0” is written<br>1 : Reset from OFD | Debug reset flag<br>0 : “0” is written<br>1 : Reset from debugger | VLTD reset flag.<br>0 : “0” is written<br>1 : Reset from VLTD | WDT reset flag<br>0 : “0” is written<br>1 : Reset from WDT | RESETpin flag<br>0 : “0” is written<br>1 : Reset from RESET pin | Power On Reset flag<br>0 : “0” is written<br>1 : Reset from Power On Reset |         |
|             |              | 15 | 14   | 13  | 12  | 11   | 10  | 9  | 8       |
| bit Symbol  |              |    |  |   |   |  |   |  |         |
| Read/Write  | R            |    |  |   |   |  |   |  |         |
| After reset | 0            | 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  | 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  | 0       |
| Function    | “0” is read. |    |  |   |   |  |   |  |         |

(Note) The TMPM370FY has power-on reset circuit and this register is initialized only by power-on reset. Therefore, “1” is set to the <PONRSTF> bit in initial reset state right after power-on.  
Note that this bit is not set by the second and subsequent resets .

## 8 Input/Output Ports

### 8.1 Port registers

- PxDATA** : **Port data register**  
To read/write port data.
- PxCR** : **Output control Register**  
To control enable/disable output.  
\* To enable/disable input, controlled 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 resistor.
- PxPDN** : **Port Pull-Down control register**  
To enable/disable pull-down resistor.
- PxIE** : **Input Enable control register**  
To enable/disable input. Default setting is disabled for avoiding through current.

## 8.2 Port Functions

### 8.2.1 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 PxlE 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 PxlE and PxCR are enabled.

The differences are summarized in the table shown below.

Table 8.1 Differences of <DRVE> setting

| Port  | I/O    | <DRVE>=0  | <DRVE>=1           |
|---|--------|---|--------------------|
| PB3,PB5<br>[When used for debug(PxFR<n>=1) and output is enabled(PxCR<n>=1)]  | Input  | x   | Depends on PxlE<n> |
|   | Output | Enable when data is valid.<br>Disable when data is invalid. |                    |
| PH0-2,PA0,PA2,PE4,PE6-7,PA7,PD3,PL0-1,PJ6-7,PK0-1<br>[When used for interrupt(PxFR<n>=1) and input is enabled(PxlE<n>=1)] | Input  | O   | O                  |
|   | Output | x   | Depends on PxCR<n> |
| Other ports   | Input  | x   | Depends on PxlE<n> |
|   | Output | x   | Depends on PxCR<n> |

O : Input or Output enabled

x : Input or Output disabled

### 8.2.2 Port A (PA0 to PA7)

The port A is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port A performs the serial interface function 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 circuit types

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

Port A register

PADATA  
(0x4000\_0000)

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

Port A control register

PACR  
(0x4000\_0004)

|             |                                     |      |      |      |      |      |      |      |
|-------------|-------------------------------------|------|------|------|------|------|------|------|
|             | 7                                   | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Bit Symbol  | PA7C                                | PA6C | PA5C | PA4C | PA3C | PA2C | PA1C | PA0C |
| Read/Write  | R/W                                 |      |      |      |      |      |      |      |
| After reset | 0                                   | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Function    | Output<br>0: disabled<br>1: enabled |      |      |      |      |      |      |      |

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<br>1:TB4IN | 0:PORT<br>1:RXD1 | 0:PORT<br>1:TXD1 | 0:PORT<br>1:SCLK1 | 0:PORT<br>1:TB1OUT | 0:PORT<br>1:TB1IN | 0:PORT<br>1:TB0OUT | 0:PORT<br>1:TB0IN |



Port A function register 2

|                         |             |                  |                   |                    |                          |              |                  |              |                  |
|-------------------------|-------------|------------------|-------------------|--------------------|--------------------------|--------------|------------------|--------------|------------------|
|                         | 7           | 6                | 5                 | 4                  | 3                        | 2            | 1                | 0            |                  |
| PAFR 2<br>(0x4000_000C) | 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            |                  |
|                         | Function    | 0:PORT<br>1:INT8 | 0:PORT<br>1:TB6IN | 0:PORT<br>1:TB6OUT | 0:PORT<br>1: <u>CTS1</u> | "0" is read. | 0:PORT<br>1:INT4 | "0" is read. | 0:PORT<br>1:INT3 |

Port A open drain Control register

|                       |             |                         |       |       |       |       |       |       |       |
|-----------------------|-------------|-------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                       | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PAOD<br>(0x4000_0028) | Bit Symbol  | PA7OD                   | PA6OD | PA5OD | PA4OD | PA3OD | PA2OD | PA1OD | PA0OD |
|                       | Read/Write  | R/W                     |       |       |       |       |       |       |       |
|                       | After reset | 0                       | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                       | Function    | 0:CMOS<br>1: open drain |       |       |       |       |       |       |       |

Port A pull-up control register

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

Port A pull-down control register

|                        |             |  |       |       |       |       |       |       |       |
|------------------------|-------------|--|-------|-------|-------|-------|-------|-------|-------|
|                        | 7           | 6                                      | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PAPDN<br>(0x4000_0030) | Bit Symbol  | PA7DN                                  | PA6DN | PA5DN | PA4DN | PA3DN | PA2DN | PA1DN | PA0DN |
|                        | Read/Write  | R/W                                    |       |       |       |       |       |       |       |
|                        | After reset | 0                                      | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                        | Function    | Pull-down<br>0: disabled<br>1: enabled |       |       |       |       |       |       |       |

Port A input enable control register

|                       |             |                                    |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                                  | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PAIE<br>(0x4000_0038) | Bit Symbol  | PA7IE                              | PA6IE | PA5IE | PA4IE | PA3IE | PA2IE | PA1IE | PA0IE |
|                       | Read/Write  | R/W                                |       |       |       |       |       |       |       |
|                       | After reset | 0                                  | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                       | Function    | Input<br>0: disabled<br>1: enabled |       |       |       |       |       |       |       |

### 8.2.3 Port B (PB0 to PB7)

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

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 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.

Port B circuit 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<br>(0x4000_0040) | Bit Symbol  | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 |
|                         | Read/Write  | R/W |     |     |     |     |     |     |     |
|                         | After reset | "0" |     |     |     |     |     |     |     |

Port B control register

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

Port B function register 1

|                        |             |                  |                 |                         |                           |                           |                            |                            |                          |
|------------------------|-------------|------------------|-----------------|-------------------------|---------------------------|---------------------------|----------------------------|----------------------------|--------------------------|
|                        | 7           | 6                | 5               | 4                       | 3                         | 2                         | 1                          | 0                          |                          |
| PBFR1<br>(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<br>1:TRST | 0:PORT<br>1:TDI | 0:PORT<br>1:TDO/<br>SWV | 0:PORT<br>1:TCK/<br>SWCLK | 0:PORT<br>1:TMS/<br>SWDIO | 0:PORT<br>1:TRACE<br>DATA1 | 0:PORT<br>1:TRACE<br>DATA0 | 0:PORT<br>1:TRACE<br>CLK |

Port B pull-up control register

|                        |             |                                      |       |       |              |                                      |       |       |       |
|------------------------|-------------|--------------------------------------|-------|-------|--------------|--------------------------------------|-------|-------|-------|
|                        | 7           | 6                                    | 5     | 4     | 3            | 2                                    | 1     | 0     |       |
| PBPUP<br>(0x4000_006C) | Bit Symbol  | PB7UP                                | PB6UP | PB5UP | -            | PB3UP                                | PB2UP | PB1UP | PB0UP |
|                        | Read/Write  | R/W                                  |       |       | R            | R/W                                  |       |       |       |
|                        | After reset | 1                                    | 1     | 0     | 0            | 1                                    | 0     | 0     | 0     |
|                        | Function    | Pull-up<br>0: disabled<br>1: enabled |       |       | "0" is read. | Pull-up<br>0: disabled<br>1: enabled |       |       |       |

Port B pull-down control register

|                        |             |              |   |   |  |              |   |   |   |
|------------------------|-------------|--------------|---|---|--|--------------|---|---|---|
|                        | 7           | 6            | 5 | 4 | 3                                      | 2            | 1 | 0 |   |
| PBPDN<br>(0x4000_0070) | Bit Symbol  | -            | - | - | PB4DN                                  | -            | - | - | - |
|                        | Read/Write  | R            |   |   | R/W                                    | R            |   |   |   |
|                        | After reset | 0            | 0 | 0 | 1                                      | 0            | 0 | 0 | 0 |
|                        | Function    | "0" is read. |   |   | Pull-down<br>0: disabled<br>1: enabled | "0" is read. |   |   |   |

Port B input enable control register

|                       |             |                                    |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                                  | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PBIE<br>(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<br>0: disabled<br>1: enabled |       |       |       |       |       |       |       |

### 8.2.4 Port C (PC0 to PC7)

The port C is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the ports C perform the input/output port for three-phase moter control (PMD).

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

Port C circuit types

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

Port C register

|             |     |     |     |     |     |     |     |     |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
|             | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit Symbol  | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| Read/Write  | R/W |     |     |     |     |     |     |     |
| After reset | "0" |     |     |     |     |     |     |     |

PCDATA  
(0x4000\_0080)

Port C control register

|             |                                     |      |      |      |      |      |      |      |
|-------------|-------------------------------------|------|------|------|------|------|------|------|
|             | 7                                   | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Bit Symbol  | PC7C                                | PC6C | PC5C | PC4C | PC3C | PC2C | PC1C | PC0C |
| Read/Write  | R/W                                 |      |      |      |      |      |      |      |
| After reset | 0                                   | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Function    | Output<br>0: disabled<br>1: enabled |      |      |      |      |      |      |      |

PCCR  
(0x4000\_0084)

Port C function register 1

|             |                  |                  |                |                |                |                |                |                |
|-------------|------------------|------------------|----------------|----------------|----------------|----------------|----------------|----------------|
|             | 7                | 6                | 5              | 4              | 3              | 2              | 1              | 0              |
| Bit Symbol  | PC7F1            | PC6F1            | PC5F1          | PC4F1          | PC3F1          | PC2F1          | PC1F1          | PC0F1          |
| Read/Write  | R/W              |                  |                |                |                |                |                |                |
| After reset | 0                | 0                | 0              | 0              | 0              | 0              | 0              | 0              |
| Function    | 0:port<br>1:OVV0 | 0:port<br>1:EMG0 | 0:port<br>1:Z0 | 0:port<br>1:W0 | 0:port<br>1:Y0 | 0:port<br>1:V0 | 0:port<br>1:X0 | 0:port<br>1:U0 |

PCFR1  
(0x4000\_0088)

Port C open drain Control register

|                       |             |                        |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       |             | 7                      | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PCOD<br>(0x4000_00A8) | Bit Symbol  | PC7OD                  | PC6OD | PC5OD | PC4OD | PC3OD | PC2OD | PC1OD | PC0OD |
|                       | Read/Write  | R/W                    |       |       |       |       |       |       |       |
|                       | After reset | 0                      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
|                       | Function    | 0:CMOS<br>1:open drain |       |       |       |       |       |       |       |

Port C pull-up control register

|                        |             |                                      |       |       |       |       |       |       |       |
|------------------------|-------------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                        |             | 7                                    | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PCPUP<br>(0x4000_00AC) | Bit Symbol  | PC7UP                                | PC6UP | PC5UP | PC4UP | PC3UP | PC2UP | PC1UP | PC0UP |
|                        | Read/Write  | R/W                                  |       |       |       |       |       |       |       |
|                        | After reset | 0                                    | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
|                        | Function    | Pull-up<br>0: disabled<br>1: enabled |       |       |       |       |       |       |       |

Port C pull-down control register

|                        |             |  |       |       |       |       |       |       |       |
|------------------------|-------------|--|-------|-------|-------|-------|-------|-------|-------|
|                        |             | 7                                      | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PCPDN<br>(0x4000_00B0) | Bit Symbol  | PC7DN                                  | PC6DN | PC5DN | PC4DN | PC3DN | PC2DN | PC1DN | PC0DN |
|                        | Read/Write  | R/W                                    |       |       |       |       |       |       |       |
|                        | After reset | 0                                      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
|                        | Function    | Pull-down<br>0: disabled<br>1: enabled |       |       |       |       |       |       |       |

Port C input enable control register

|                       |             |                                  |       |       |       |       |       |       |       |
|-----------------------|-------------|----------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       |             | 7                                | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PCIE<br>(0x4000_00B8) | Bit Symbol  | PC7IE                            | PC6IE | PC5IE | PC4IE | PC3IE | PC2IE | PC1IE | PC0IE |
|                       | Read/Write  | R/W                              |       |       |       |       |       |       |       |
|                       | After reset | 0                                | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
|                       | Function    | Input<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

### 8.2.5 Port D (PD0 to PD6)

The port D is a general-purpose, 7-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port D performs the serial interface function, 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)** 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 D circuit types

|      |   |    |    |    |    |    |     |     |
|------|---|----|----|----|----|----|-----|-----|
|      | 7 | 6  | 5  | 4  | 3  | 2  | 1   | 0   |
| Type | - | T3 | T2 | T9 | T4 | T3 | T10 | T11 |

Port D register

|                         |             |              |     |     |     |     |     |     |     |
|-------------------------|-------------|--------------|-----|-----|-----|-----|-----|-----|-----|
|                         | 7           | 6            | 5   | 4   | 3   | 2   | 1   | 0   |     |
| PDDATA<br>(0x4000_00C0) | Bit Symbol  | -            | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
|                         | Read/Write  | R            | R/W |     |     |     |     |     |     |
|                         | After reset | "0" is read. | "0" |     |     |     |     |     |     |

Port D control register

|                       |             |              |                                     |      |      |      |      |      |      |
|-----------------------|-------------|--------------|-------------------------------------|------|------|------|------|------|------|
|                       | 7           | 6            | 5                                   | 4    | 3    | 2    | 1    | 0    |      |
| PDCR<br>(0x4000_00C4) | Bit Symbol  | -            | PD6C                                | PD5C | PD4C | PD3C | PD2C | PD1C | PD0C |
|                       | Read/Write  | R            | R/W                                 |      |      |      |      |      |      |
|                       | After reset | 0            | 0                                   | 0    | 0    | 0    | 0    | 0    | 0    |
|                       | Function    | "0" is read. | Output<br>0: disabled<br>1: enabled |      |      |      |      |      |      |

Port D function register 1

|                        |             |              |                  |                  |                   |                  |                   |                   |                   |
|------------------------|-------------|--------------|------------------|------------------|-------------------|------------------|-------------------|-------------------|-------------------|
|                        | 7           | 6            | 5                | 4                | 3                 | 2                | 1                 | 0                 |                   |
| PDFR1<br>(0x4000_00C8) | Bit Symbol  | -            | PD6F1            | PD5F1            | PD4F1             | PD3F1            | PD2F1             | PD1F1             | PD0F1             |
|                        | Read/Write  | R            | R/W              |                  |                   |                  |                   |                   |                   |
|                        | After reset | 0            | 0                | 0                | 0                 | 0                | 0                 | 0                 | 0                 |
|                        | Function    | "0" is read. | 0:PORT<br>1:RXD2 | 0:PORT<br>1:TXD2 | 0:PORT<br>1:SCLK2 | 0:PORT<br>1:INT9 | 0:PORT<br>1:ENCZ0 | 0:PORT<br>1:ENCB0 | 0:PORT<br>1:ENCA0 |

Port D function register 2

|                         |             |              |   |   |                  |              |   |                                |
|-------------------------|-------------|--------------|---|---|------------------|--------------|---|--------------------------------|
|                         | 7           | 6            | 5 | 4 | 3                | 2            | 1 | 0                              |
| PDFR 2<br>(0x4000_00CC) | Bit Symbol  | -            | - | - | PD4F2            | -            | - | PD1F2 PD0F2                    |
|                         | Read/Write  | R            |   |   | R/W              | R            |   | R/W R/W                        |
|                         | After reset | 0            |   |   | 0                | 0            |   | 0 0                            |
|                         | Function    | "0" is read. |   |   | 0:PORT<br>1:CTS2 | "0" is read. |   | 0:PORT<br>1:TB5OUT<br>1: TB5IN |

Port D open drain control register

|                       |             |              |                         |       |       |       |       |             |  |
|-----------------------|-------------|--------------|-------------------------|-------|-------|-------|-------|-------------|--|
|                       | 7           | 6            | 5                       | 4     | 3     | 2     | 1     | 0           |  |
| PDOD<br>(0x4000_00E8) | Bit Symbol  | -            | PD6OD                   | PD5OD | PD4OD | PD3OD | PD2OD | PD1OD PD0OD |  |
|                       | Read/Write  | R            | R/W                     |       |       |       |       |             |  |
|                       | After reset | 0            | 0                       | 0     | 0     | 0     | 0     | 0 0         |  |
|                       | Function    | "0" is read. | 0:CMOS<br>1: open drain |       |       |       |       |             |  |

Port D pull-up control register

|                        |             |              |                                    |       |       |       |       |             |  |
|------------------------|-------------|--------------|------------------------------------|-------|-------|-------|-------|-------------|--|
|                        | 7           | 6            | 5                                  | 4     | 3     | 2     | 1     | 0           |  |
| PDPUP<br>(0x4000_00EC) | Bit Symbol  | -            | PD6UP                              | PD5UP | PD4UP | PD3UP | PD2UP | PD1UP PD0UP |  |
|                        | Read/Write  | R            | R/W                                |       |       |       |       |             |  |
|                        | After reset | 0            | 0                                  | 0     | 0     | 0     | 0     | 0 0         |  |
|                        | Function    | "0" is read. | Pull-up<br>0:disabled<br>1:enabled |       |       |       |       |             |  |

Port D pull-down control register

|                        |             |              |                                      |       |       |       |       |             |  |
|------------------------|-------------|--------------|--------------------------------------|-------|-------|-------|-------|-------------|--|
|                        | 7           | 6            | 5                                    | 4     | 3     | 2     | 1     | 0           |  |
| PDPDN<br>(0x4000_00F0) | Bit Symbol  | -            | PD6DN                                | PD5DN | PD4DN | PD3DN | PD2DN | PD1DN PD0DN |  |
|                        | Read/Write  | R            | R/W                                  |       |       |       |       |             |  |
|                        | After reset | 0            | 0                                    | 0     | 0     | 0     | 0     | 0 0         |  |
|                        | Function    | "0" is read. | Pull-down<br>0:disabled<br>1:enabled |       |       |       |       |             |  |

Port D input enable control register

|                       |             |              |                                  |       |       |       |       |             |  |
|-----------------------|-------------|--------------|----------------------------------|-------|-------|-------|-------|-------------|--|
|                       | 7           | 6            | 5                                | 4     | 3     | 2     | 1     | 0           |  |
| PDIE<br>(0x4000_00F8) | Bit Symbol  | -            | PD6IE                            | PD5IE | PD4IE | PD3IE | PD2IE | PD1IE PD0IE |  |
|                       | Read/Write  | R            | R/W                              |       |       |       |       |             |  |
|                       | After reset | 0            | 0                                | 0     | 0     | 0     | 0     | 0 0         |  |
|                       | Function    | "0" is read. | Input<br>0:disabled<br>1:enabled |       |       |       |       |             |  |

### 8.2.6 Port E (PE0 to PE7)

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

|             |     |     |     |     |     |     |     |     |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
|             | 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" |     |     |     |     |     |     |     |

PEDATA  
(0x4000\_0100)

Port E control register

|             |                                     |      |      |      |      |      |      |      |
|-------------|-------------------------------------|------|------|------|------|------|------|------|
|             | 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<br>0: disabled<br>1: enabled |      |      |      |      |      |      |      |

PECR  
(0x4000\_0104)

Port E functions register 1

|             |                     |                   |                    |                   |                    |                   |                  |                  |
|-------------|---------------------|-------------------|--------------------|-------------------|--------------------|-------------------|------------------|------------------|
|             | 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<br>1:TB3OUT. | 0:PORT<br>1:TB3IN | 0:PORT<br>1:TB2OUT | 0:PORT<br>1:TB2IN | 0:PORT<br>1:TB4OUT | 0:PORT<br>1:SCLK0 | 0:PORT<br>1:RXD0 | 0:PORT<br>1:TXD0 |

PEFR1  
(0x4000\_0108)



Port E functions register 2

|                        |             |                   |                  |              |                  |              |                  |              |
|------------------------|-------------|-------------------|------------------|--------------|------------------|--------------|------------------|--------------|
|                        | 7           | 6                 | 5                | 4            | 3                | 2            | 1                | 0            |
| PEFR2<br>(0x4000_010C) | 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<br>1:INT7. | 0:PORT<br>1:INT6 | "0" is read. | 0:PORT<br>1:INT5 | "0" is read. | 0:PORT<br>1:CTS0 | "0" is read. |

Port E open drain control register

|                       |             |                        |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                      | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PEOD<br>(0x4000_0128) | Bit Symbol  | PE7OD                  | PE6OD | PE5OD | PE4OD | PE3OD | PE2OD | PE1OD |       |
|                       | Read/Write  | R/W                    |       |       |       |       |       |       | PE0OD |
|                       | After reset | 0                      | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                       | Function    | 0:CMOS<br>1:Open drain |       |       |       |       |       |       |       |

Port E pull-up control register

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

Port E pull-down control register

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

Port E input enable control register

|                       |             |                                  |       |       |       |       |       |       |       |
|-----------------------|-------------|----------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                                | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PEIE<br>(0x4000_0138) | Bit Symbol  | PE7IE                            | PE6IE | PE5IE | PE4IE | PE3IE | PE2IE | PE1IE |       |
|                       | Read/Write  | R/W                              |       |       |       |       |       |       | PE0IE |
|                       | After reset | 0                                | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                       | Function    | Input<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

### 8.2.7 Port F (PF0 to PF4)

The port F is a general-purpose, 5-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port F performs the 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”.

Reset initializes all bits of the port F as general-purpose ports with input, output and pull-down disabled. Pull-up is enabled for PF0 and disabled for PF1 through PF4.

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   |
| Bit Symbol  | -            | - | - | PF4 | PF3 | PF2 | PF1 | PF0 |
| Read/Write  | R            |   |   | R/W |     |     |     |     |
| After reset | “0” is read. |   |   | “0” |     |     |     |     |

PFDATA  
(0x4000\_0140)

Port F control register

|             |              |   |   |                                     |      |      |      |      |
|-------------|--------------|---|---|-------------------------------------|------|------|------|------|
|             | 7            | 6 | 5 | 4                                   | 3    | 2    | 1    | 0    |
| Bit Symbol  | -            | - | - | PF4C                                | PF3C | PF2C | PF1C | PF0C |
| Read/Write  | R            |   |   | R/W                                 |      |      |      |      |
| After reset | 0            |   |   | 0                                   | 0    | 0    | 0    | 0    |
| Function    | “0” is read. |   |   | Output<br>0: disabled<br>1: enabled |      |      |      |      |

PF0C  
(0x4000\_0144)

Port F function register 1

|             |              |   |   |                    |                    |                    |                    |                    |
|-------------|--------------|---|---|--------------------|--------------------|--------------------|--------------------|--------------------|
|             | 7            | 6 | 5 | 4                  | 3                  | 2                  | 1                  | 0                  |
| Bit Symbol  | -            | - | - | PF4F1              | PF3F1              | PF2F1              | PF1F1              | PF0F1              |
| Read/Write  | R            |   |   | R/W                |                    |                    |                    |                    |
| After reset | 0            |   |   | 0                  | 0                  | 0                  | 0                  | 0                  |
| Function    | “0” is read. |   |   | 0:PORT<br>1: ENCZ1 | 0:PORT<br>1: ENCB1 | 0:PORT<br>1: ENCA1 | 0:PORT<br>1:TB7OUT | 0:PORT<br>1: TB7IN |

PFFR1  
(0x4000\_0148)

Port F function register 2

|             |              |   |   |                  |                  |                   |              |   |
|-------------|--------------|---|---|------------------|------------------|-------------------|--------------|---|
|             | 7            | 6 | 5 | 4                | 3                | 2                 | 1            | 0 |
| Bit Symbol  | -            | - | - | PF4F2            | PF3F2            | PF2F2             | -            | - |
| Read/Write  | R            |   |   | R/W              |                  |                   | R            |   |
| After reset | 0            |   |   | 0                | 0                | 0                 | 0            |   |
| Function    | “0” is read. |   |   | 0:PORT<br>1:RXD3 | 0:PORT<br>1:TXD3 | 0:PORT<br>1:SCLK3 | “0” is read. |   |

PFFR2  
(0x4000\_014C)

Port F function register 3

|             |              |   |   |   |   |                  |              |   |
|-------------|--------------|---|---|---|---|------------------|--------------|---|
|             | 7            | 6 | 5 | 4 | 3 | 2                | 1            | 0 |
| Bit Symbol  | -            | - | - | - | - | PF2F3            | -            | - |
| Read/Write  | R            |   |   |   |   | R/W              | R            |   |
| After reset | 0            |   |   |   |   | 0                | 0            |   |
| Function    | "0" is read. |   |   |   |   | 0:PORT<br>1:CTS3 | "0" is read. |   |

PFFR3  
(0x4000\_0150)

Port F open drain control register

|             |             |   |   |                        |       |       |       |       |
|-------------|-------------|---|---|------------------------|-------|-------|-------|-------|
|             | 7           | 6 | 5 | 4                      | 3     | 2     | 1     | 0     |
| 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<br>1:Open drain |       |       |       |       |

PFOOD  
(0x4000\_0168)

Port F pull-up control register

|             |             |   |   |                                    |       |       |       |       |
|-------------|-------------|---|---|------------------------------------|-------|-------|-------|-------|
|             | 7           | 6 | 5 | 4                                  | 3     | 2     | 1     | 0     |
| 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<br>0:disabled<br>1:enabled |       |       |       |       |

PFPUP  
(0x4000\_016C)

Port F pull-down control register

|             |             |   |   |                                      |       |       |       |       |
|-------------|-------------|---|---|--------------------------------------|-------|-------|-------|-------|
|             | 7           | 6 | 5 | 4                                    | 3     | 2     | 1     | 0     |
| 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<br>0:disabled<br>1:enabled |       |       |       |       |

PFPDN  
(0x4000\_0170)

Port F input enable control register

|             |             |   |   |                                  |       |       |       |       |
|-------------|-------------|---|---|----------------------------------|-------|-------|-------|-------|
|             | 7           | 6 | 5 | 4                                | 3     | 2     | 1     | 0     |
| Bit Symbol  | -           | - | - | PF4IE                            | PF3IE | PF2IE | PF1IE | PF0IE |
| Read/Write  | R           |   |   | R/W                              |       |       |       |       |
| After reset | 0           |   |   | 0                                | 0     | 0     | 0     | 0     |
| Function    | "0" is read |   |   | Input<br>0:disabled<br>1:enabled |       |       |       |       |

PFIE  
(0x4000\_0178)

### 8.2.8 Port G (PG0 to PG7)

The port G is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port G performs the input/output port for three-phase moter 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<br>(0x4000_0180) | PG7 | PG6 | PG5 | PG4 | PG3 | PG2 | PG1 | PG0 |
| Read/Write              | R/W |     |     |     |     |     |     |     |
| After reset             | "0" |     |     |     |     |     |     |     |

Port G control register

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

Port G function register 1

|                        | 7                | 6                | 5               | 4               | 3               | 2               | 1               | 0               |
|------------------------|------------------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| PGFR1<br>(0x4000_0188) | PG7F1            | PG6F1            | PG5F1           | PG4F1           | PG3F1           | PG2F1           | PG1F1           | PG0F1           |
| Read/Write             | R/W              |                  |                 |                 |                 |                 |                 |                 |
| After reset            | 0                | 0                | 0               | 0               | 0               | 0               | 0               | 0               |
| Function               | 0:PORT<br>1: OV1 | 0:PORT<br>1: EM1 | 0:PORT<br>1: Z1 | 0:PORT<br>1: W1 | 0:PORT<br>1: Y1 | 0:PORT<br>1: V1 | 0:PORT<br>1: X1 | 0:PORT<br>1: U1 |

Port G open drain control register

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

Port G pull-up control register

|                        |             |                                    |       |       |       |       |       |       |       |  |
|------------------------|-------------|------------------------------------|-------|-------|-------|-------|-------|-------|-------|--|
|                        | 7           | 6                                  | 5     | 4     | 3     | 2     | 1     | 0     |       |  |
| PGPUP<br>(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<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |  |

Port G pull-down control register

|                        |             |                                      |       |       |       |       |       |       |       |  |
|------------------------|-------------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|--|
|                        | 7           | 6                                    | 5     | 4     | 3     | 2     | 1     | 0     |       |  |
| PGPDN<br>(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<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |  |

Port G input enable control register

|                       |             |                                  |       |       |       |       |       |       |       |  |
|-----------------------|-------------|----------------------------------|-------|-------|-------|-------|-------|-------|-------|--|
|                       | 7           | 6                                | 5     | 4     | 3     | 2     | 1     | 0     |       |  |
| PGIE<br>(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<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |  |

### 8.2.9 Port H (PH0 to PH7)

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

Reset initializes all bits of the port H 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 PHFR1 register and enable input in the PHIE 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 H as analog input pins, conversion accuracy may be reduced. Be sure to verify that this causes no problem on your system.

Port H circuit type

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

Port H register

|                         |     |     |     |     |     |     |     |     |
|-------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
|                         | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| PHDATA<br>(0x4000_01C0) | PH7 | PH6 | PH5 | PH4 | PH3 | PH2 | PH1 | PH0 |
| Read/Write              | R/W |     |     |     |     |     |     |     |
| After reset             | "0" |     |     |     |     |     |     |     |

Port H control register

|                       |                                     |      |      |      |      |      |      |      |
|-----------------------|-------------------------------------|------|------|------|------|------|------|------|
|                       | 7                                   | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PHCR<br>(0x4000_01C4) | PH7C                                | PH6C | PH5C | PH4C | PH3C | PH2C | PH1C | PH0C |
| Read/Write            | R/W                                 |      |      |      |      |      |      |      |
| After reset           | 0                                   | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Function              | Output<br>0: disabled<br>1: enabled |      |      |      |      |      |      |      |

Port H function register 1

|                        |             |   |   |   |   |                  |                  |                  |
|------------------------|-------------|---|---|---|---|------------------|------------------|------------------|
|                        | 7           | 6 | 5 | 4 | 3 | 2                | 1                | 0                |
| PHFR1<br>(0x4000_01C8) | -           | - | - | - | - | PH2F1            | PH1F1            | PH0F1            |
| Read/Write             | R           |   |   |   |   | R/W              |                  |                  |
| After reset            | 0           | 0 | 0 | 0 | 0 | 0                | 0                | 0                |
| Function               | "0" is read |   |   |   |   | 0:PORT<br>1:INT2 | 0:PORT<br>1:INT1 | 0:PORT<br>1:INT0 |

Port H open drain control register

|                       |             |                        |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                      | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PHOD<br>(0x4000_01E8) | Bit Symbol  | PH7OD                  | PH6OD | PH5OD | PH4OD | PH3OD | PH2OD | PH1OD | PH0OD |
|                       | Read/Write  | R/W                    |       |       |       |       |       |       |       |
|                       | After reset | 0                      | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                       | Function    | 0:CMOS<br>1:open drain |       |       |       |       |       |       |       |

Port H pull-up control register

|                        |             |                                    |       |       |       |       |       |       |       |
|------------------------|-------------|------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                        | 7           | 6                                  | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PHPUP<br>(0x4000_01EC) | Bit Symbol  | PH7UP                              | PH6UP | PH5UP | PH4UP | PH3UP | PH2UP | PH1UP | PH0UP |
|                        | Read/Write  | R/W                                |       |       |       |       |       |       |       |
|                        | After reset | 0                                  | 0     | 0     | 0     | 0     | 0     | 1     |       |
|                        | Function    | Pull-up<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

Port H pull-down control register

|                        |             |                                      |       |       |       |       |       |       |       |
|------------------------|-------------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                        | 7           | 6                                    | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PHPDN<br>(0x4000_01F0) | Bit Symbol  | PH7DN                                | PH6DN | PH5DN | PH4DN | PH3DN | PH2DN | PH1DN | PH0DN |
|                        | Read/Write  | R/W                                  |       |       |       |       |       |       |       |
|                        | After reset | 0                                    | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                        | Function    | Pull-down<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

Port H input enable control register

|                       |             |                                  |       |       |       |       |       |       |       |
|-----------------------|-------------|----------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       | 7           | 6                                | 5     | 4     | 3     | 2     | 1     | 0     |       |
| PHIE<br>(0x4000_01F8) | Bit Symbol  | PH7IE                            | PH6IE | PH5IE | PH4IE | PH3IE | PH2IE | PH1IE | PH0IE |
|                       | Read/Write  | R/W                              |       |       |       |       |       |       |       |
|                       | After reset | 0                                | 0     | 0     | 0     | 0     | 0     | 0     |       |
|                       | Function    | Input<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

8.2.10 Port I (PI0 to PI3)

The port I is a general-purpose, 4-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port 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 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 | T16 | T16 | T16 |

Port I register

PIDATA  
(0x4000\_0200)

|             |             |   |   |   |     |     |     |     |
|-------------|-------------|---|---|---|-----|-----|-----|-----|
|             | 7           | 6 | 5 | 4 | 3   | 2   | 1   | 0   |
| Bit Symbol  | -           | - | - | - | PI3 | PI2 | PI1 | PI0 |
| Read/Write  | R           |   |   |   | R/W |     |     |     |
| After reset | "0" is read |   |   |   | "0" |     |     |     |

Port I control register

PICR  
(0x4000\_0204)

|             |             |   |   |   |                                     |      |      |      |
|-------------|-------------|---|---|---|-------------------------------------|------|------|------|
|             | 7           | 6 | 5 | 4 | 3                                   | 2    | 1    | 0    |
| Bit Symbol  | -           | - | - | - | PI3C                                | PI2C | PI1C | PI0C |
| Read/Write  | R           |   |   |   | R/W                                 |      |      |      |
| After reset | 0           | 0 | 0 | 0 | 0                                   | 0    | 0    | 0    |
| Function    | "0" is read |   |   |   | Output<br>0: disabled<br>1: enabled |      |      |      |

Port I open drain control register

PIOD  
(0x4000\_0228)

|             |             |   |   |   |                        |       |       |       |
|-------------|-------------|---|---|---|------------------------|-------|-------|-------|
|             | 7           | 6 | 5 | 4 | 3                      | 2     | 1     | 0     |
| Bit Symbol  | -           | - | - | - | PI3OD                  | PI2OD | PI1OD | PI0OD |
| Read/Write  | R           |   |   |   | R/W                    |       |       |       |
| After reset | 0           | 0 | 0 | 0 | 0                      | 0     | 0     | 0     |
| Function    | "0" is read |   |   |   | 0:CMOS<br>1:open drain |       |       |       |

Port I pull-up control register

PIPUP  
(0x4000\_022C)

|             |             |   |   |   |                                      |       |       |       |
|-------------|-------------|---|---|---|--------------------------------------|-------|-------|-------|
|             | 7           | 6 | 5 | 4 | 3                                    | 2     | 1     | 0     |
| Bit Symbol  | -           | - | - | - | PI3UP                                | PI2UP | PI1UP | PI0UP |
| Read/Write  | R           |   |   |   | R/W                                  |       |       |       |
| After reset | 0           | 0 | 0 | 0 | 0                                    | 0     | 0     | 0     |
| Function    | "0" is read |   |   |   | Pull-up<br>0: disabled<br>1: enabled |       |       |       |



Port I pull-down control register

|             |             |   |   |   |                                      |       |       |       |
|-------------|-------------|---|---|---|--------------------------------------|-------|-------|-------|
|             | 7           | 6 | 5 | 4 | 3                                    | 2     | 1     | 0     |
| Bit Symbol  | -           | - | - | - | PI3DN                                | PI2DN | PI1DN | PI0DN |
| Read/Write  | R           |   |   |   | R/W                                  |       |       |       |
| After reset | 0           | 0 | 0 | 0 | 0                                    | 0     | 0     | 0     |
| Function    | "0" is read |   |   |   | Pull-down<br>0:disabled<br>1:enabled |       |       |       |

PIPDN  
(0x4000\_0230)

Port I input enable control register

|             |             |   |   |   |                                  |       |       |       |
|-------------|-------------|---|---|---|----------------------------------|-------|-------|-------|
|             | 7           | 6 | 5 | 4 | 3                                | 2     | 1     | 0     |
| Bit Symbol  | -           | - | - | - | PI3IE                            | PI2IE | PI1IE | PI0IE |
| Read/Write  | R           |   |   |   | R/W                              |       |       |       |
| After reset | 0           | 0 | 0 | 0 | 0                                | 0     | 0     | 0     |
| Function    | "0" is read |   |   |   | Input<br>0:disabled<br>1:enabled |       |       |       |

PIIE  
(0x4000\_0238)

### 8.2.11 Port J (PJ0 to PJ7)

The port J is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port J performs the analog input of the 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 J 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   |
| Bit Symbol  | PJ7 | PJ6 | PJ5 | PJ4 | PJ3 | PJ2 | PJ1 | PJ0 |
| Read/Write  | R/W |     |     |     |     |     |     |     |
| After reset | "0" |     |     |     |     |     |     |     |

PJDATA  
(0x4000\_0240)

Port J control register

|             |                                     |      |      |      |      |      |      |      |
|-------------|-------------------------------------|------|------|------|------|------|------|------|
|             | 7                                   | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Bit Symbol  | PJ7C                                | PJ6C | PJ5C | PJ4C | PJ3C | PJ2C | PJ1C | PJ0C |
| Read/Write  | R/W                                 |      |      |      |      |      |      |      |
| After reset | 0                                   | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Function    | Output<br>0: disabled<br>1: enabled |      |      |      |      |      |      |      |

PJCR  
(0x4000\_0244)

Port J function register 1

|             |                  |                  |             |   |   |   |   |   |
|-------------|------------------|------------------|-------------|---|---|---|---|---|
|             | 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<br>1:INTD | 0:PORT<br>1:INTC | "0" is read |   |   |   |   |   |

PJFR1  
(0x4000\_0248)

Port J open drain control register

|                       |             |                        |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       |             | 7                      | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PJOD<br>(0x4000_0268) | 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<br>1:open drain |       |       |       |       |       |       |       |

Port J pull-up control register

|                       |             |                                    |       |       |       |       |       |       |       |
|-----------------------|-------------|------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       |             | 7                                  | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PJUP<br>(0x4000_026C) | 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<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

Port J pull-down control register

|                        |             |                                      |       |       |       |       |       |       |       |
|------------------------|-------------|--------------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                        |             | 7                                    | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PJPDN<br>(0x4000_0270) | 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<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

Port J input enable control register

|                       |             |                                  |       |       |       |       |       |       |       |
|-----------------------|-------------|----------------------------------|-------|-------|-------|-------|-------|-------|-------|
|                       |             | 7                                | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PJIE<br>(0x4000_0278) | 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<br>0:disabled<br>1:enabled |       |       |       |       |       |       |       |

### 8.2.12 Port K (PK0 to PK1)

The port K is a general-purpose, 2-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port K performs the 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 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<br>(0x4000_0280) | -           | - | - | - | - | - | PK1 | PK0 |
| Read/Write              | R           |   |   |   |   |   | R/W |     |
| After reset             | "0" is read |   |   |   |   |   | "0" |     |

Port K control register

|                       | 7            | 6 | 5 | 4 | 3 | 2 | 1                                   | 0    |
|-----------------------|--------------|---|---|---|---|---|-------------------------------------|------|
| PKCR<br>(0x4000_0284) | -            | - | - | - | - | - | PK1C                                | PK0C |
| Read/Write            | R            |   |   |   |   |   | R/W                                 |      |
| After reset           | 0            |   |   |   |   |   | 0                                   | 0    |
| Function              | "0" is read. |   |   |   |   |   | Output<br>0: disabled<br>1: enabled |      |

Port K function register 1

|                        | 7           | 6 | 5 | 4 | 3 | 2 | 1                | 0                |
|------------------------|-------------|---|---|---|---|---|------------------|------------------|
| PKFR1<br>(0x4000_0288) | -           | - | - | - | - | - | PK1F1            | PK0F1            |
| Read/Write             | R           |   |   |   |   |   | R/W              |                  |
| After reset            | 0           |   |   |   |   |   | 0                | 0                |
| Function               | "0" is read |   |   |   |   |   | 0:PORT<br>1:INTF | 0:PORT<br>1:INTE |

Port K open drain control register

|                       |             |             |   |   |   |   |                        |             |  |
|-----------------------|-------------|-------------|---|---|---|---|------------------------|-------------|--|
|                       | 7           | 6           | 5 | 4 | 3 | 2 | 1                      | 0           |  |
| PKOD<br>(0x4000_02A8) | Bit Symbol  | -           | - | - | - | - | -                      | PK1OD PK0OD |  |
|                       | Read/Write  | R           |   |   |   |   | R/W                    |             |  |
|                       | After reset | 0           | 0 | 0 | 0 | 0 | 0                      | 0           |  |
|                       | Function    | "0" is read |   |   |   |   | 0:CMOS<br>1:open drain |             |  |

Port K pull-up control register

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

Port K pull-down control register

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

Port K input enable control register

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

8.2.13 Port L(PL0 to PL1)

The port L is a general-purpose, 2-bit input port. For this port, inputs can be specified in units of bits. Besides the general-purpose input function, the port L performs the functions as the external interrupt input.

Reset initializes all bits of the port L as general-purpose input ports with input disabled.

To use the external interrupt input for releasing STOP mode, select this function in the PLFR1 register and enable input in the PLIE 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.

**(Note1)** 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.

**(Note2)** When the power supply switch on, please keep 'L' level to port\_L, constant time. (include in reset time) The details please watch 'Notice for the power supply' of 'Electrical Characteristics'.

Port L circuit type

|      |   |   |   |   |   |   |    |    |
|------|---|---|---|---|---|---|----|----|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1  | 0  |
| Type | - | - | - | - | - | - | T5 | T5 |

Port L register

|                     |             |   |   |   |   |   |     |     |
|---------------------|-------------|---|---|---|---|---|-----|-----|
|                     | 7           | 6 | 5 | 4 | 3 | 2 | 1   | 0   |
| PL<br>(0x4000_02C0) | -           | - | - | - | - | - | PL1 | PL0 |
| Read/Write          | R           |   |   |   |   |   | R   |     |
| After reset         | "0" is read |   |   |   |   |   | "0" |     |

Port L function register 1

|                        |             |   |   |   |   |   |                   |                   |
|------------------------|-------------|---|---|---|---|---|-------------------|-------------------|
|                        | 7           | 6 | 5 | 4 | 3 | 2 | 1                 | 0                 |
| PLFR1<br>(0x4000_02C8) | -           | - | - | - | - | - | PL1F1             | PL0F1             |
| Read/Write             | R           |   |   |   |   |   | R/W               |                   |
| After reset            | 0           |   |   |   |   |   | 0                 | 0                 |
| Function               | "0" is read |   |   |   |   |   | 0:PORT<br>1: INTA | 0:PORT<br>1: INTB |

Port L input enable control register

|                       |             |   |   |   |   |   |                                  |       |
|-----------------------|-------------|---|---|---|---|---|----------------------------------|-------|
|                       | 7           | 6 | 5 | 4 | 3 | 2 | 1                                | 0     |
| PLIE<br>(0x4000_02F8) | -           | - | - | - | - | - | PL1IE                            | PL0IE |
| Read/Write            | R           |   |   |   |   |   | R/W                              |       |
| After reset           | 0           |   |   |   |   |   | 0                                | 0     |
| Function              | "0" is read |   |   |   |   |   | Input<br>0:disabled<br>1:enabled |       |

## 8.3 Block Diagrams of Ports

### 8.3.1 Port Types

The ports are classified into 21 types shown below. Please refer to the following pages for the block diagrams of each port type.

Table 8.2 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     | -         | -  | Function output triggered by enable signal |
| T7   | i/o  | i         | -         | -         | -      | NoR     | -         | -  |  |
| T8   | i/o  | i         | -         | -         | -      | -       | 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/o       | -         | -      | 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       | -         | -  |  |
| T19  | i/o  | o         | -         | -         | -      | NoR     | -         | -  | Function output triggered by enable signal |
| T20  | i/o  | i         | -         | -         | -      | NoR     | NoR       | ○  | BOOT input enabled during reset            |

R : Forced disable during reset.

NoR : Unaffected by reset.

### 8.3.2 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.

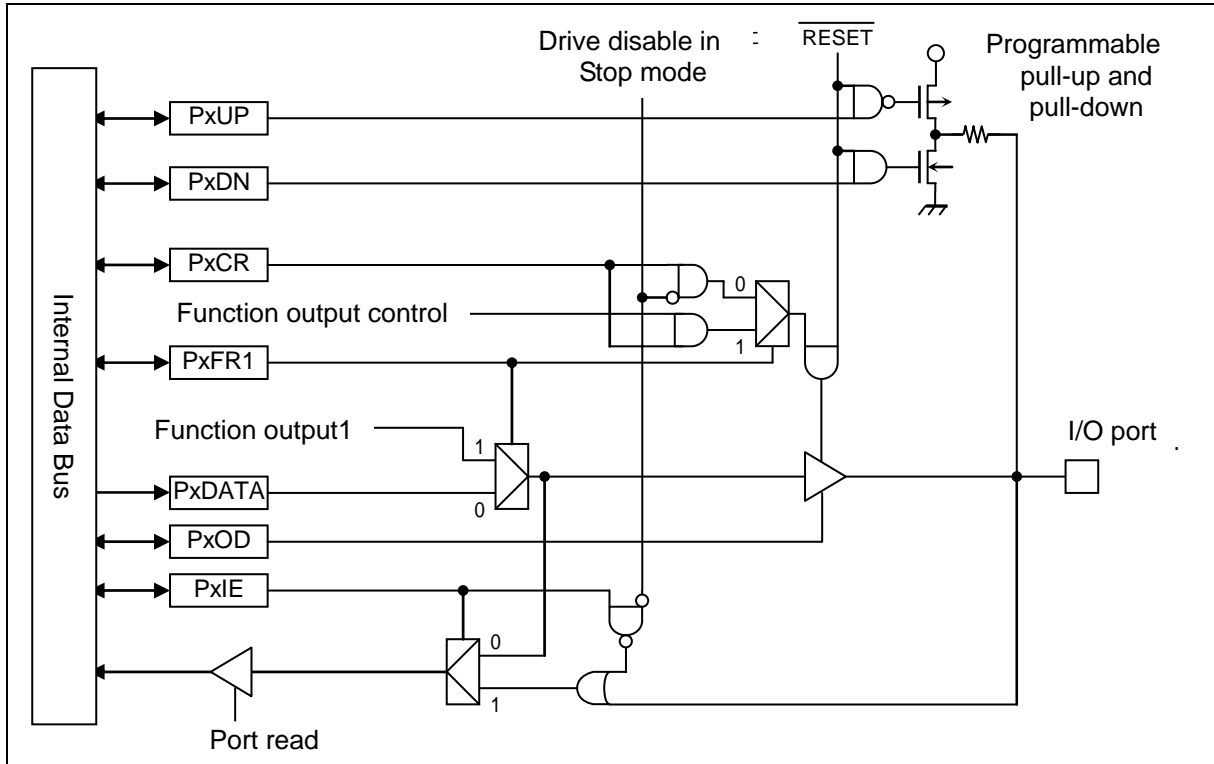


Fig 8.1 Type T1



### 8.3.3 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.

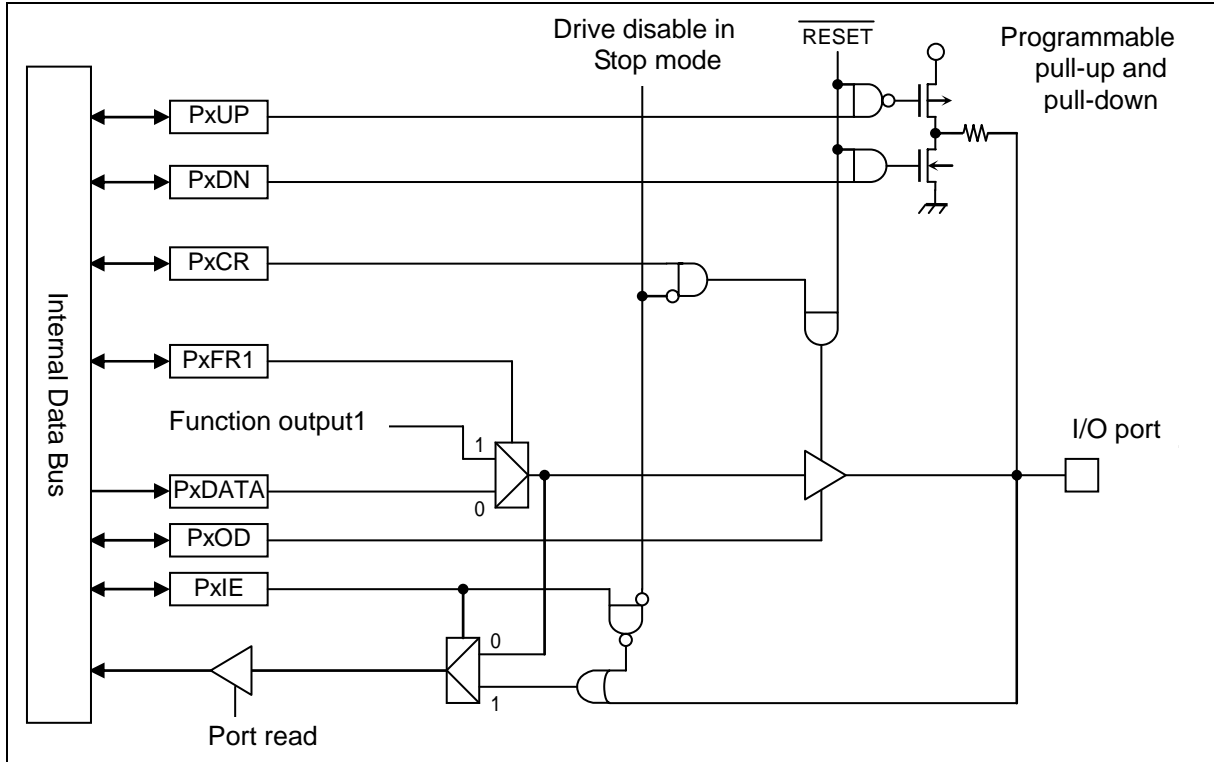


Fig 8.2 Type T2

### 8.3.4 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.

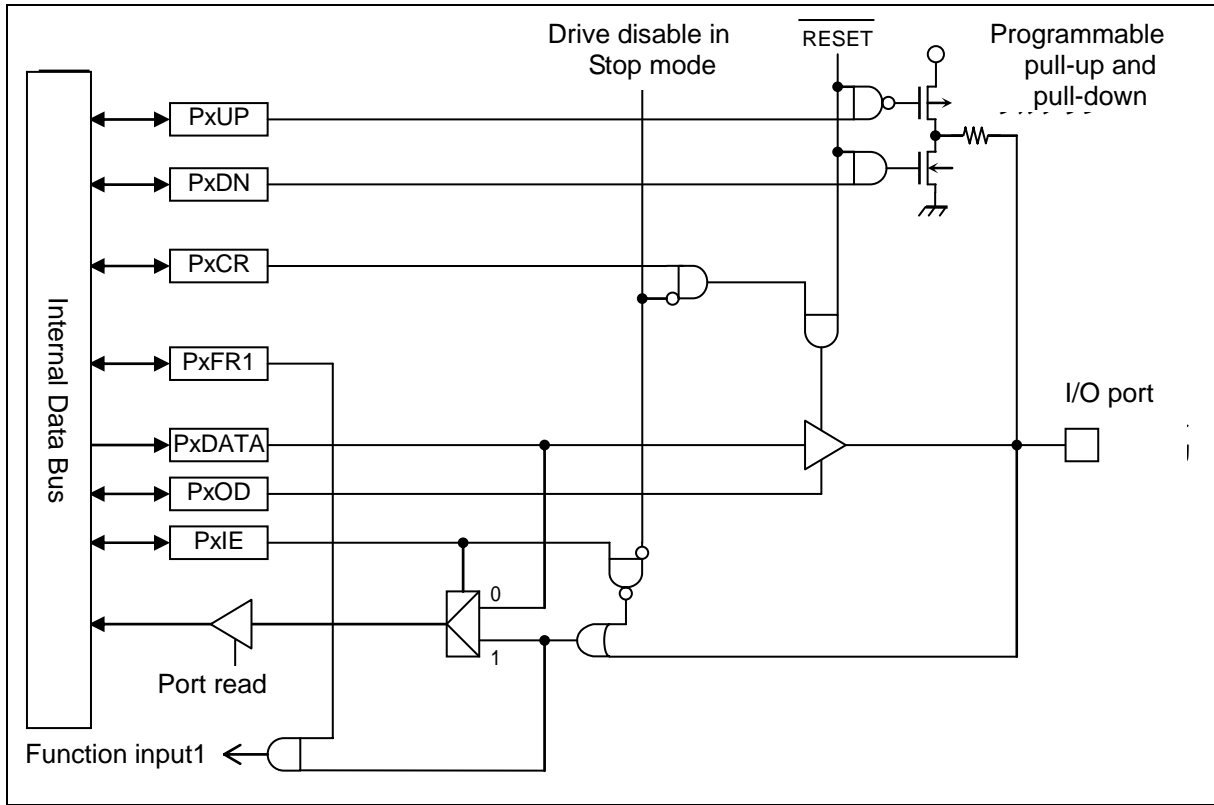


Fig 8.3 Type T3

### 8.3.5 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.

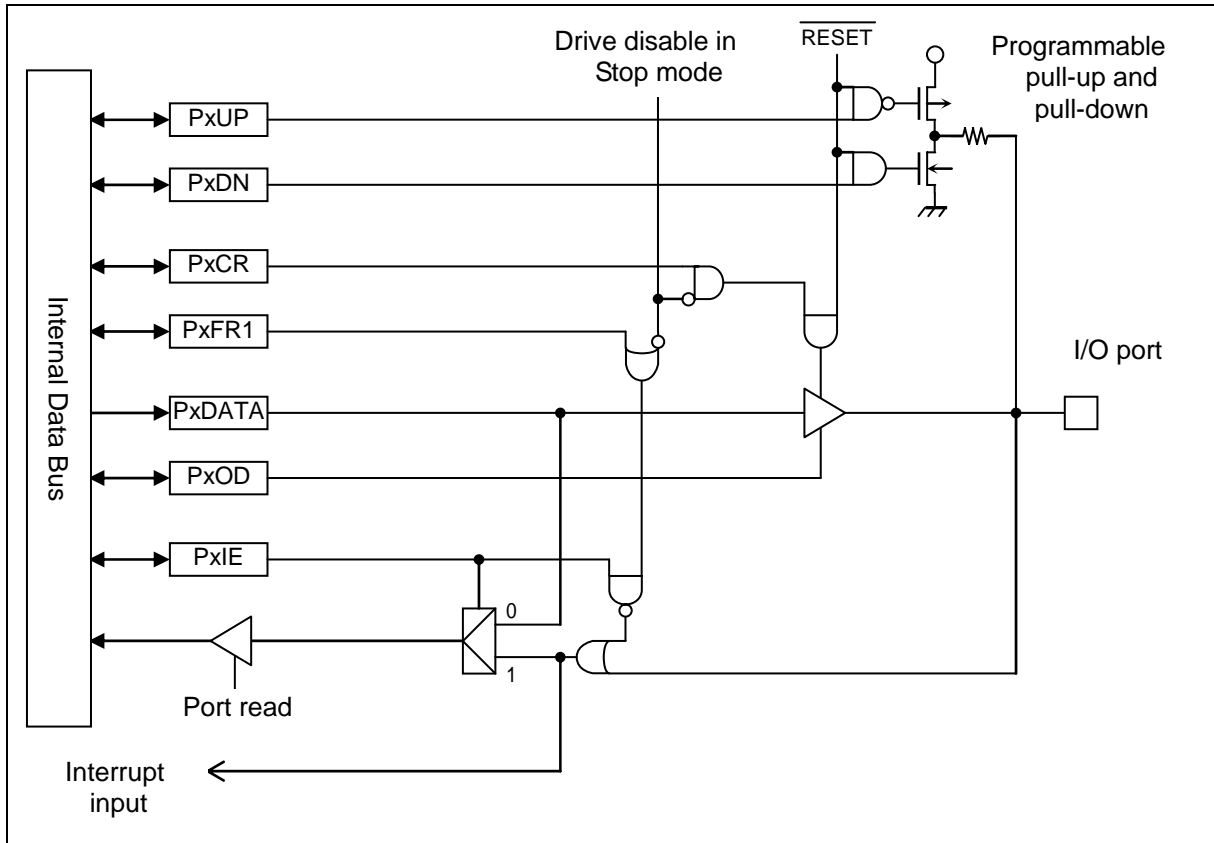


Fig 8.4 Type T4

### 8.3.6 Type T5

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

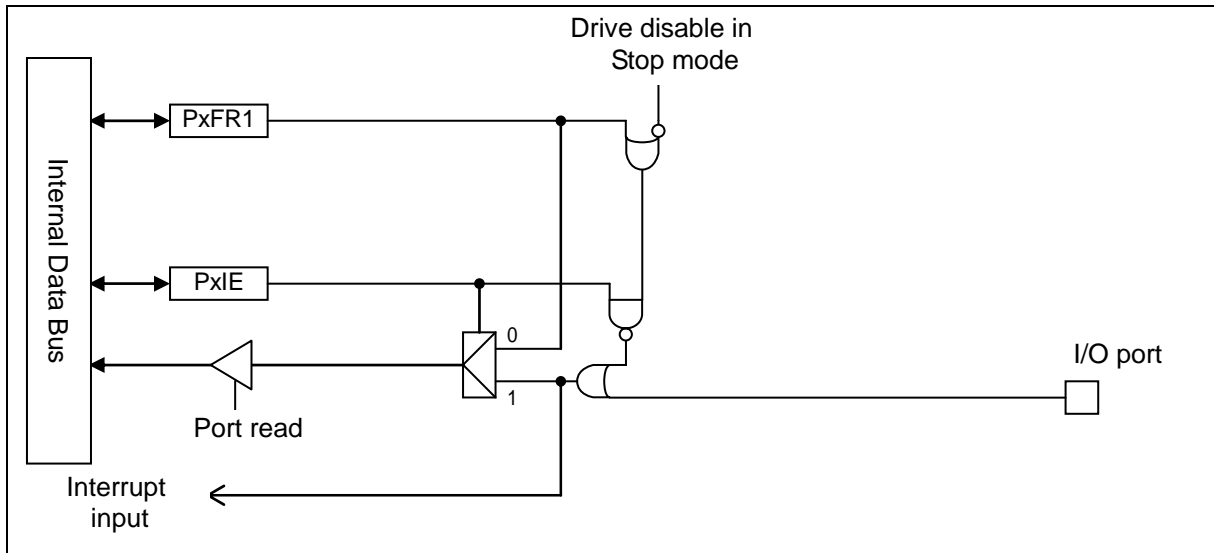


Fig 8.5 Type T5

### 8.3.7 Type T6

Type T6 is a general-purpose port with pull-up. 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.

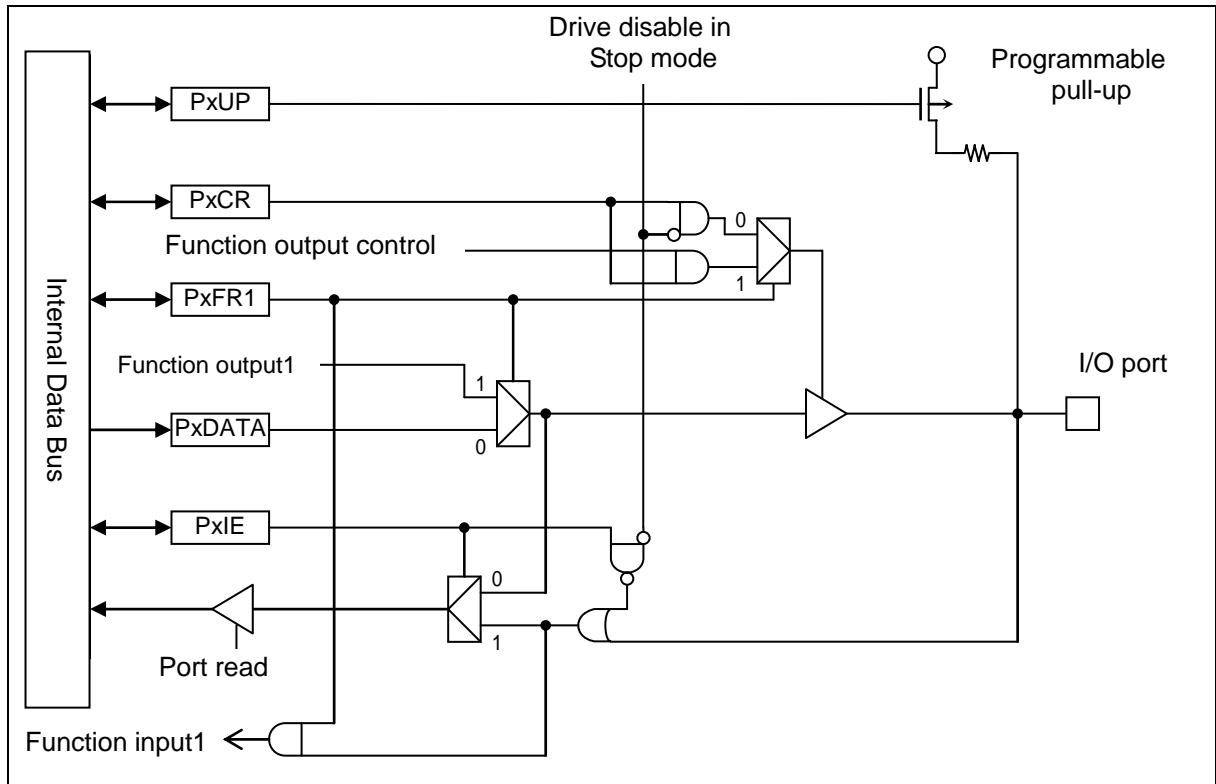


Fig 8.6 Type T6

### 8.3.8 Type T7

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

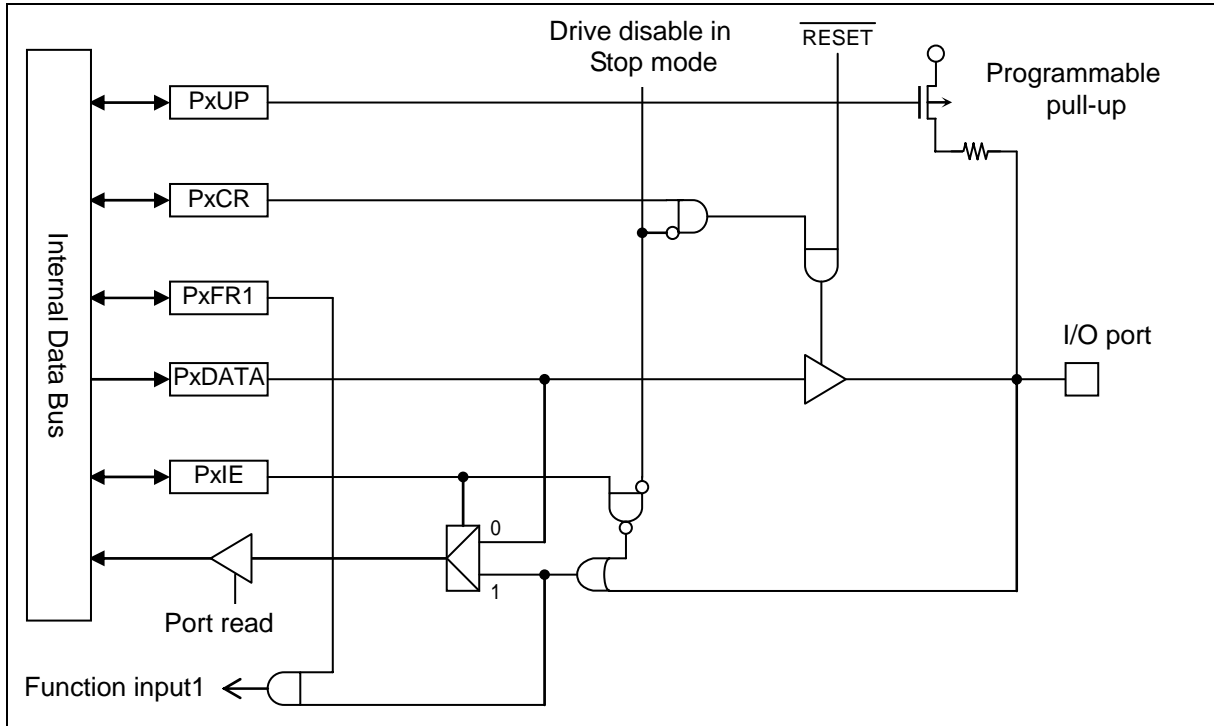


Fig 8.7 Type T7

### 8.3.9 Type T8

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

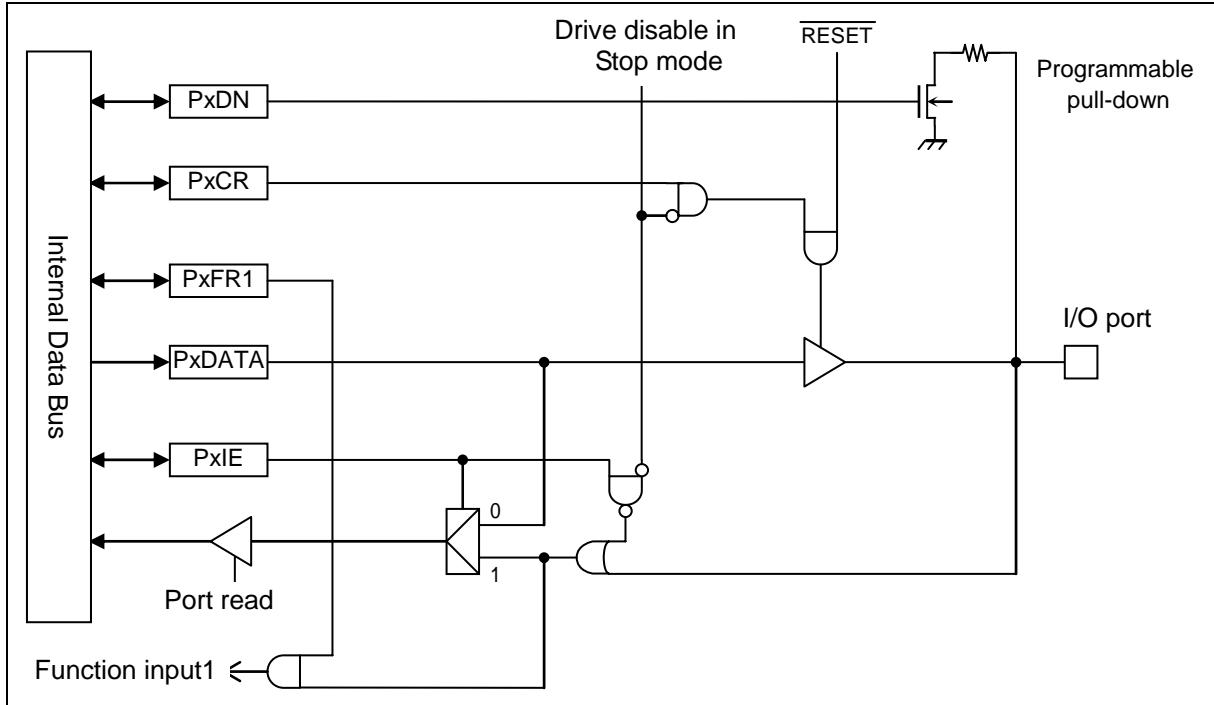


Fig 8.8 Type T8

8.3.10 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.

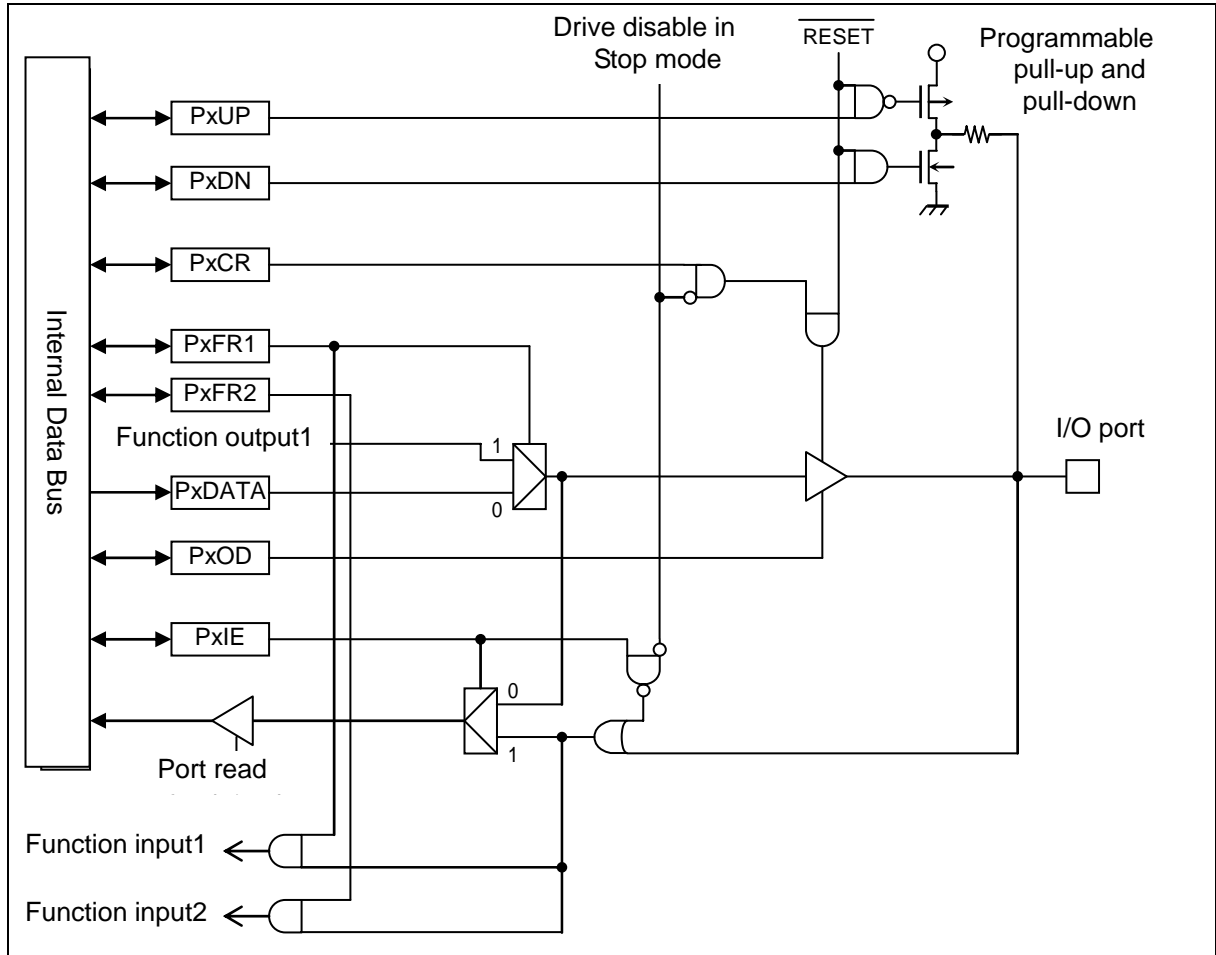


Fig 8.9 Type T9



### 8.3.11 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.

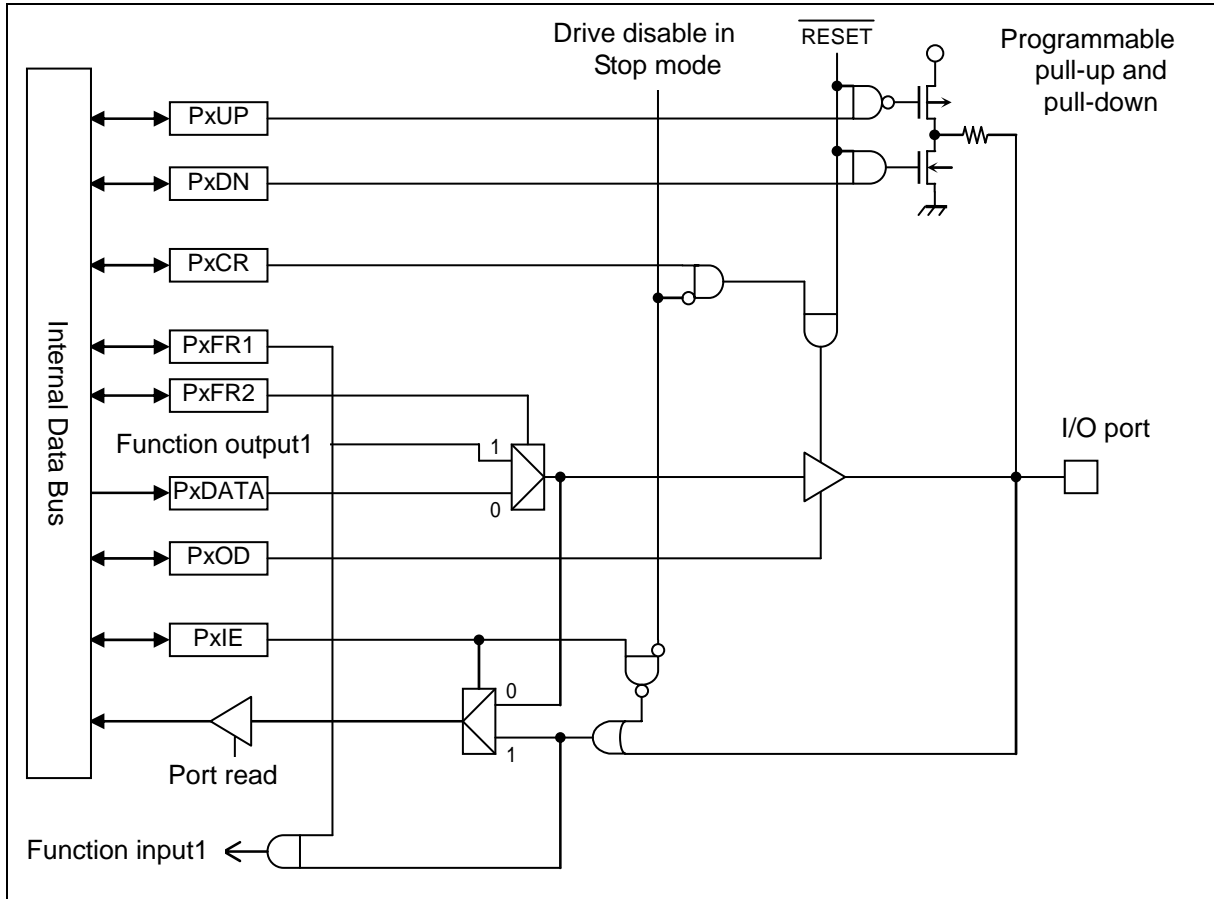


Fig 8.10 Type T10

### 8.3.12 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.

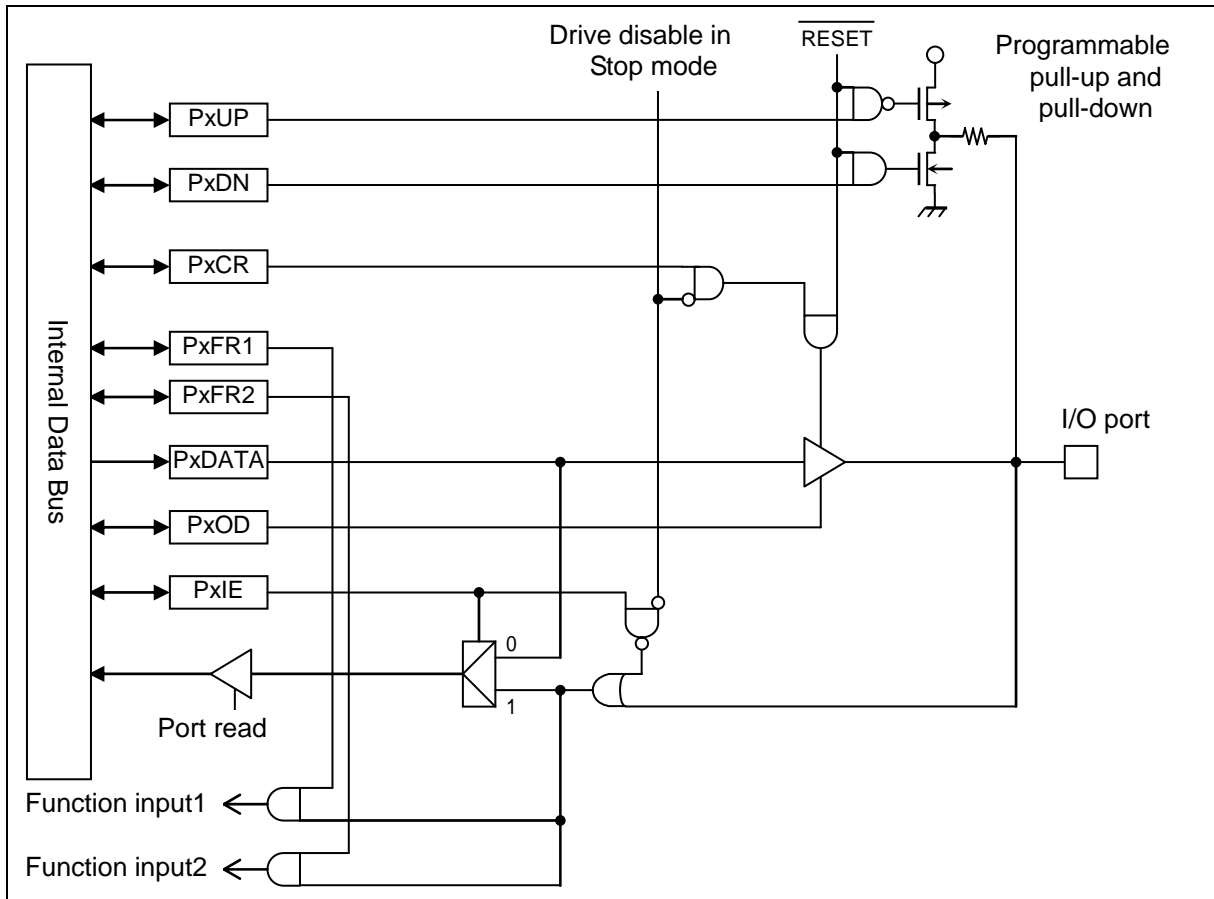


Fig 8.11 Type T11

### 8.3.13 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.

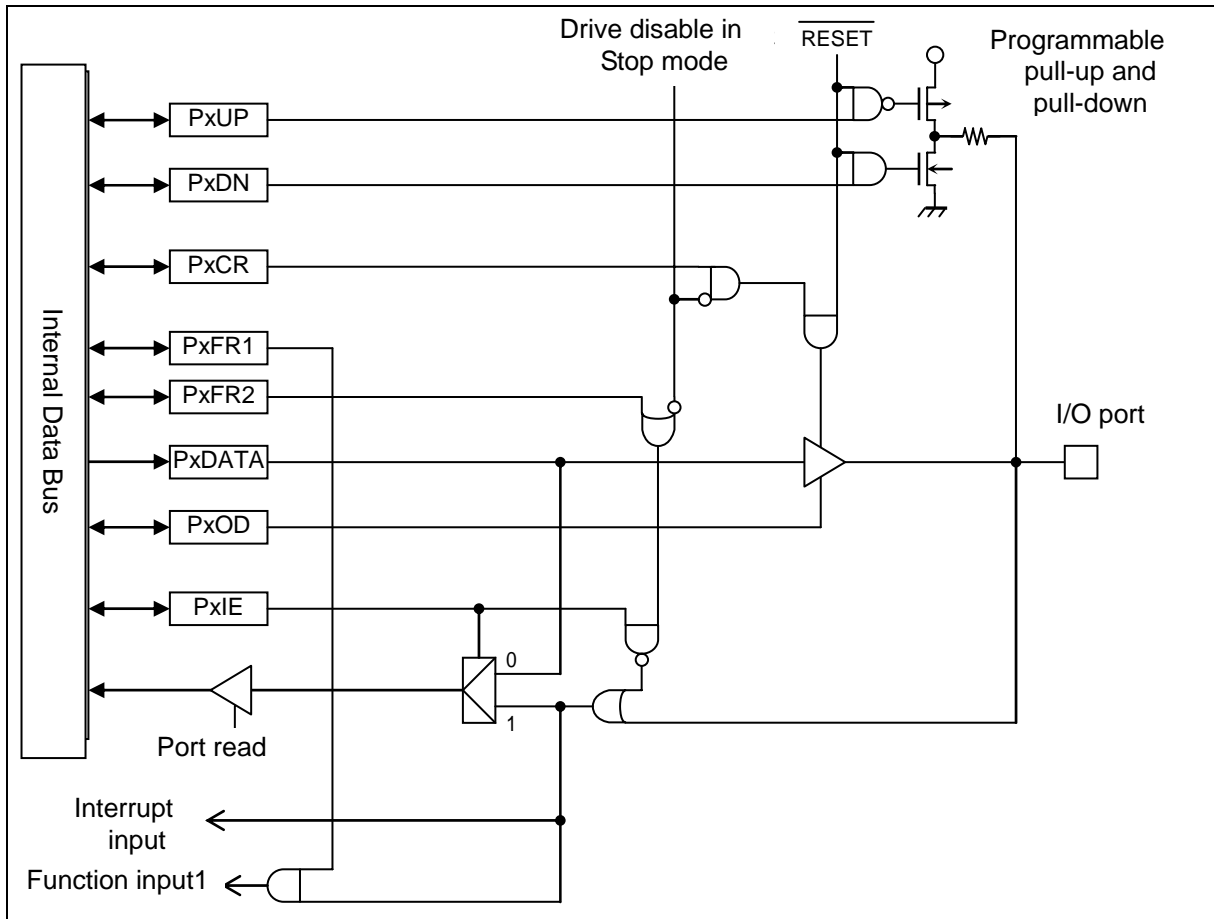


Fig 8.12 Type T12

### 8.3.14 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.

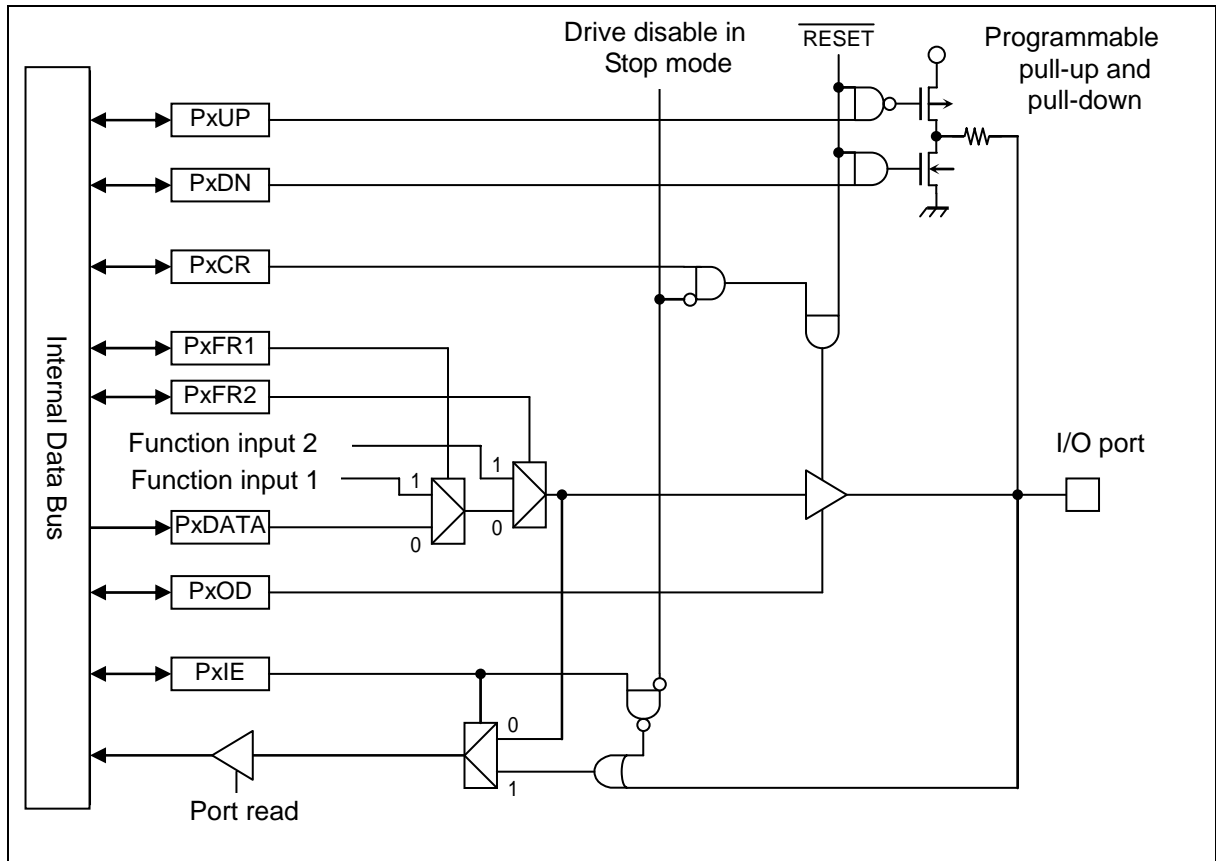


Fig 8.13 Type T13

### 8.3.15 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.

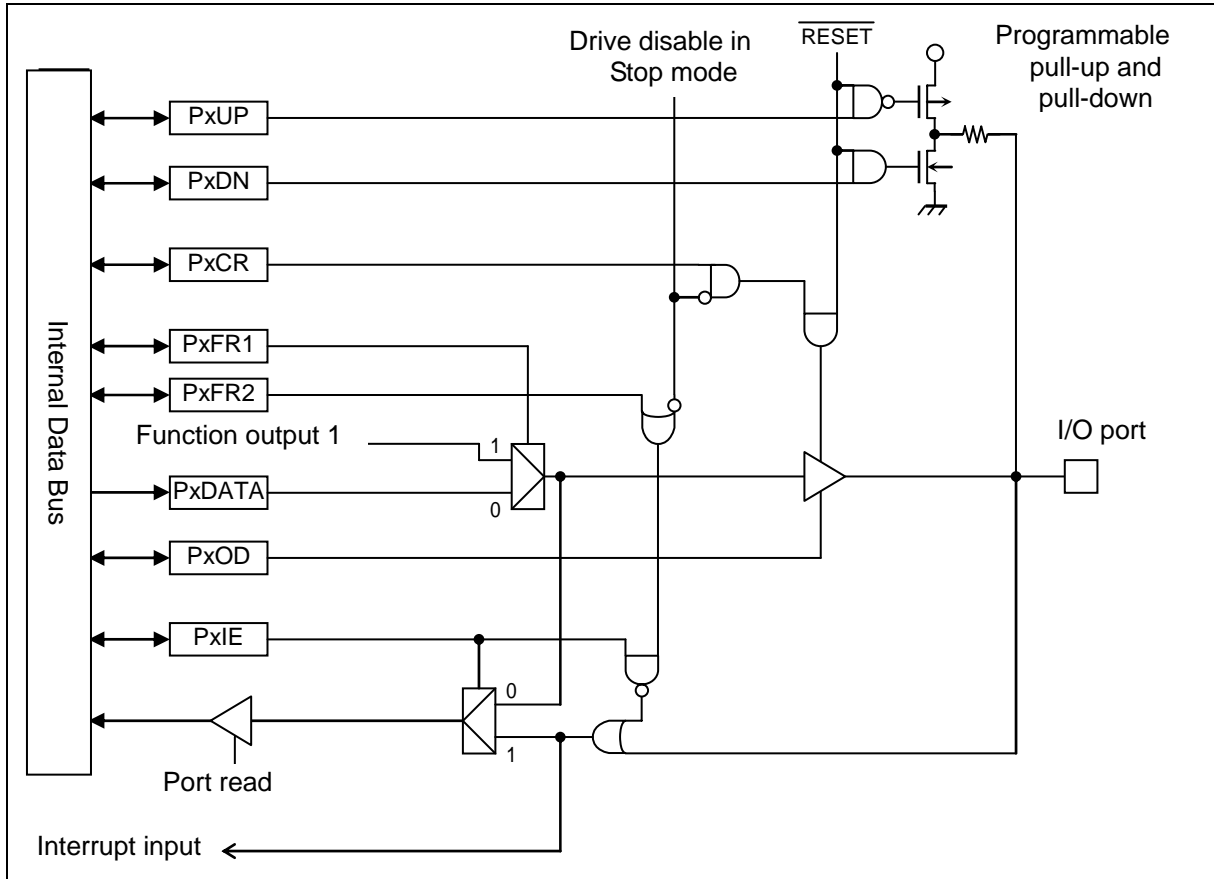


Fig 8.14 Type T14

8.3.16 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.

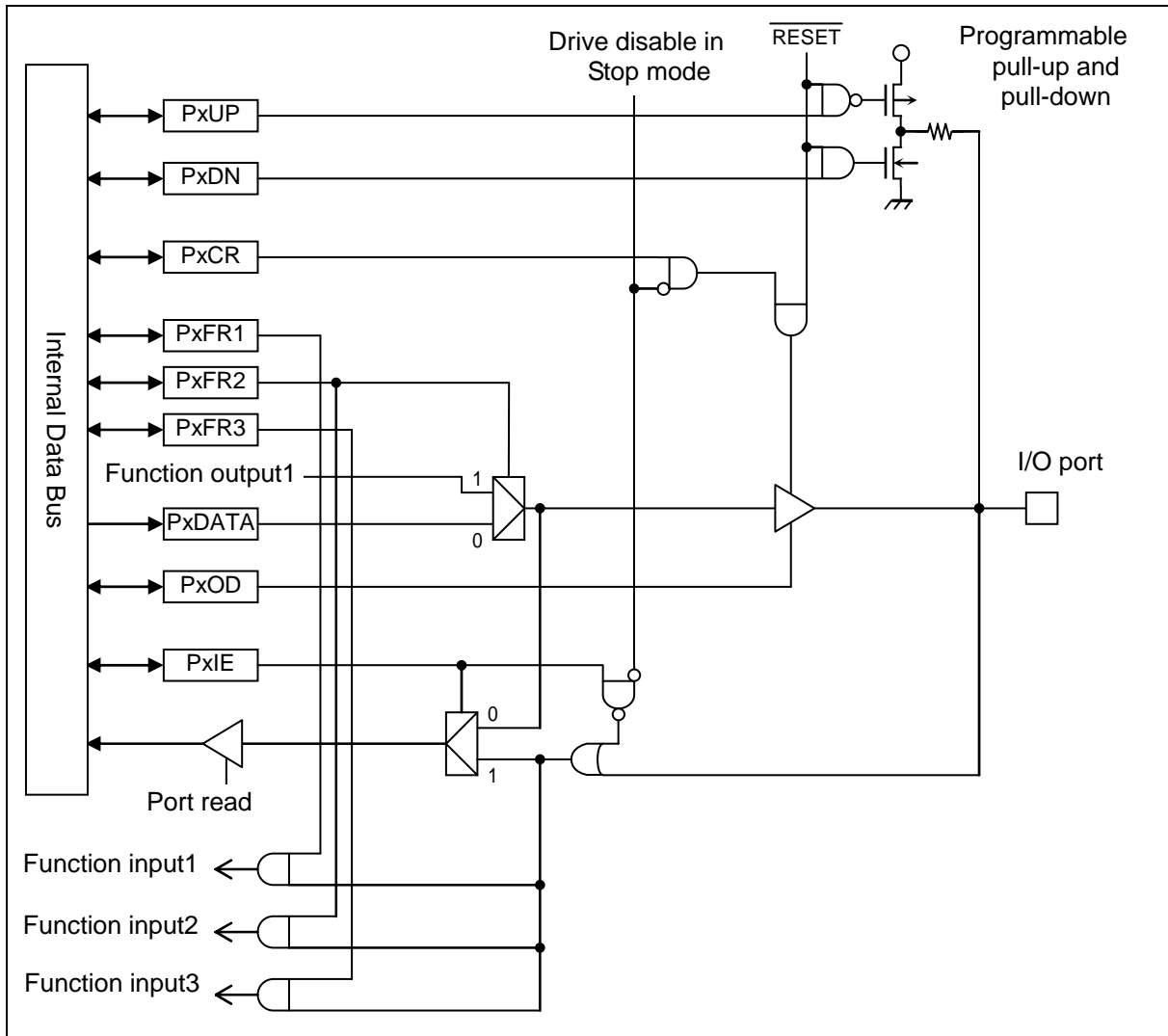


Fig 8.15 Type T15

### 8.3.17 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.

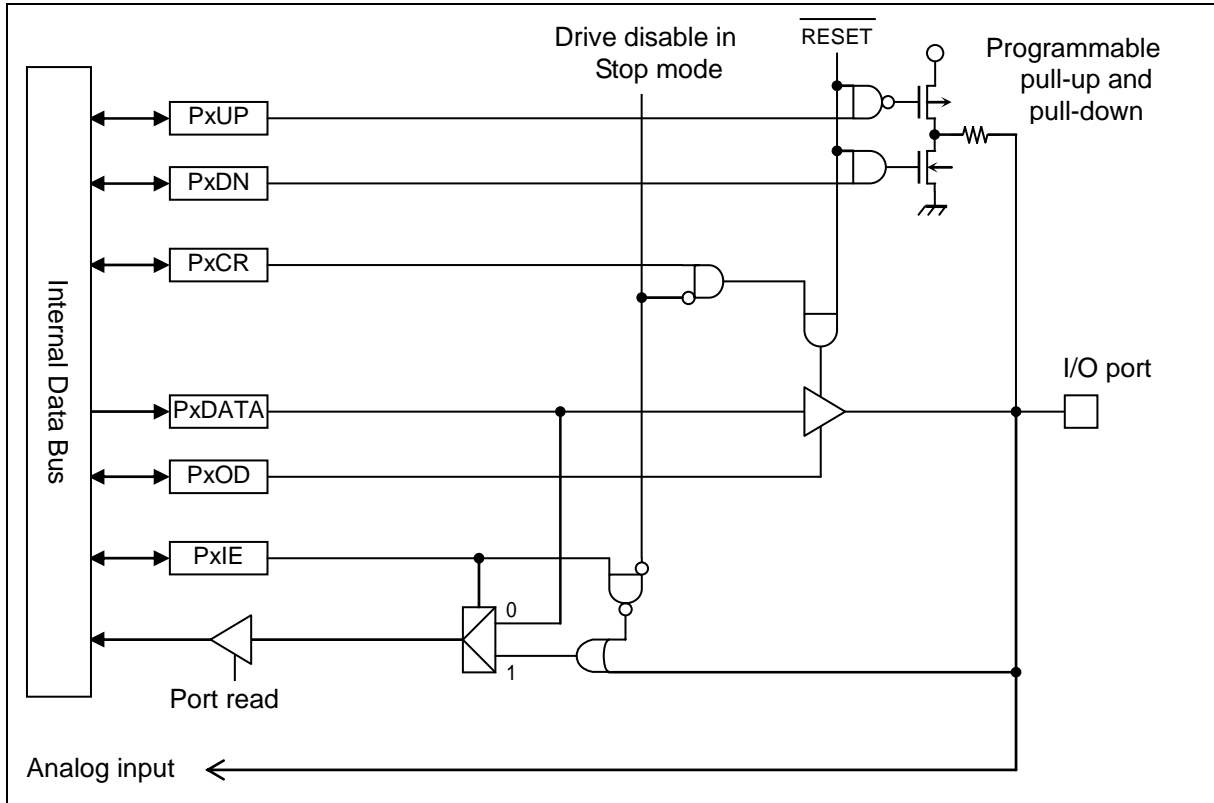


Fig 8.16 Type T16

### 8.3.18 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.

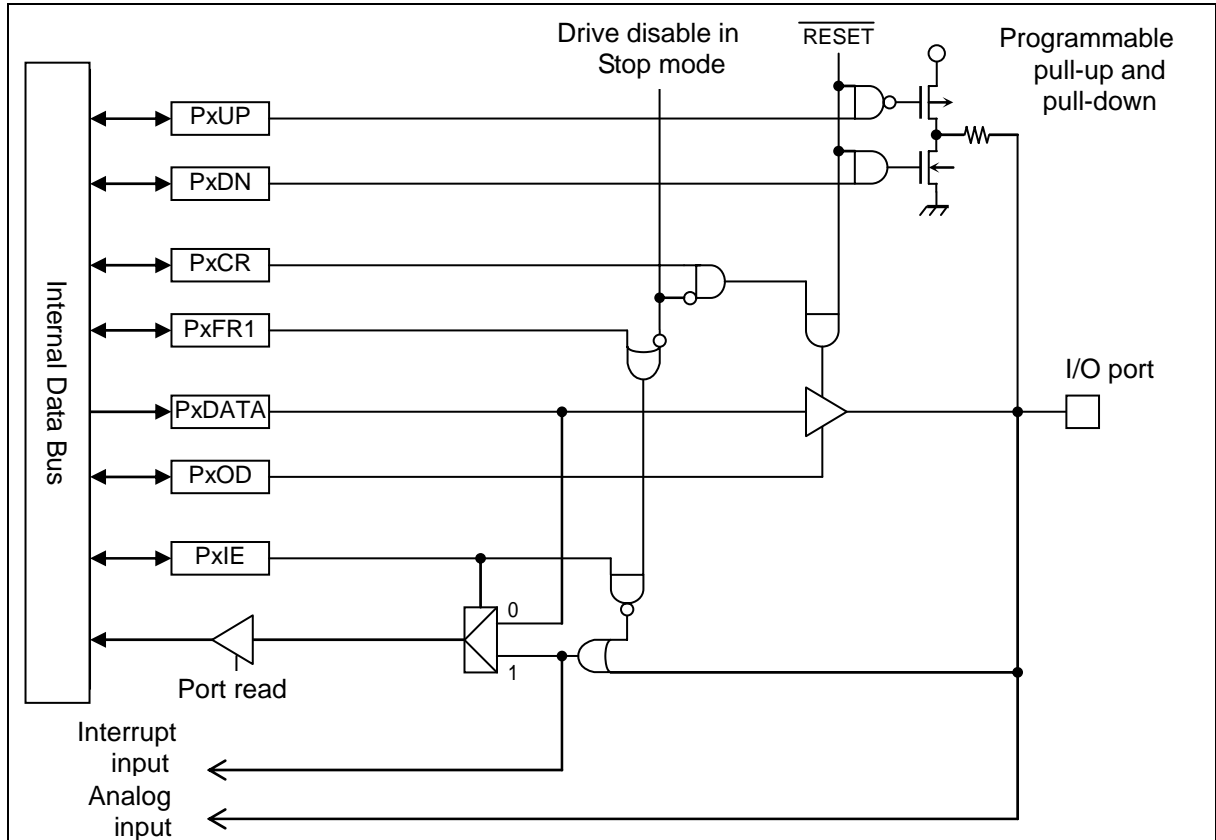


Fig 8.17 Type T17



8.3.19 Type T18

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

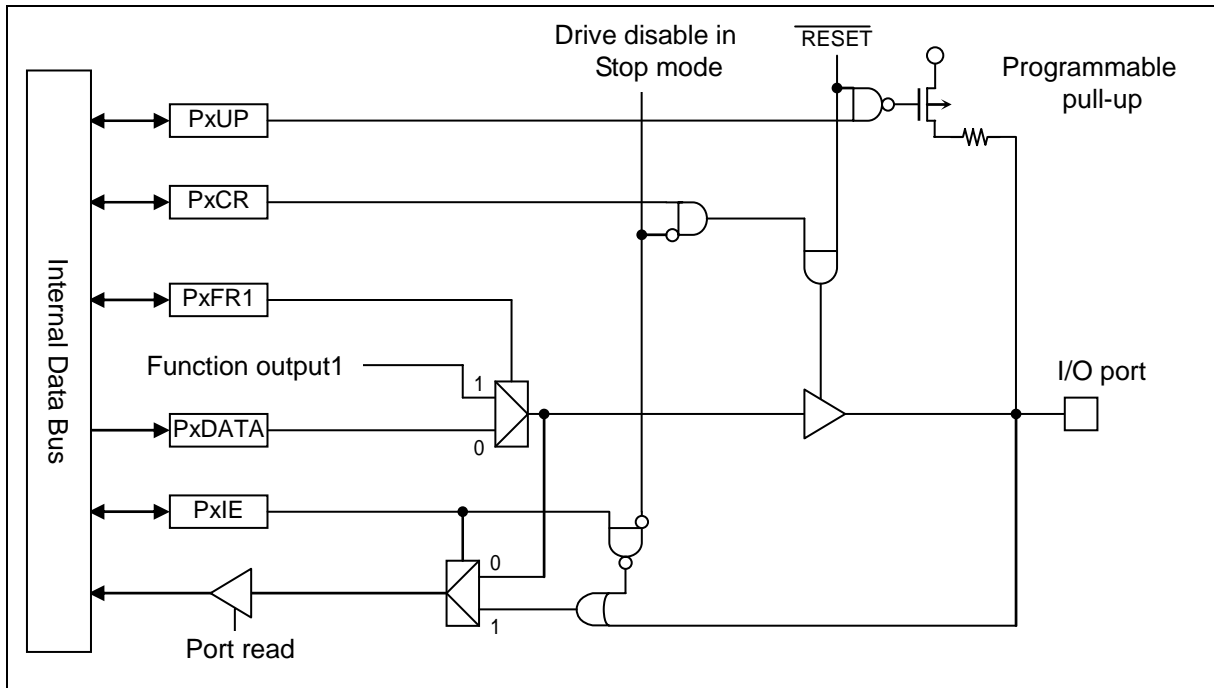


Fig 8.18 Type T18

8.3.20 Type T19

Type T19 is a general-purpose port with pull-up. 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.

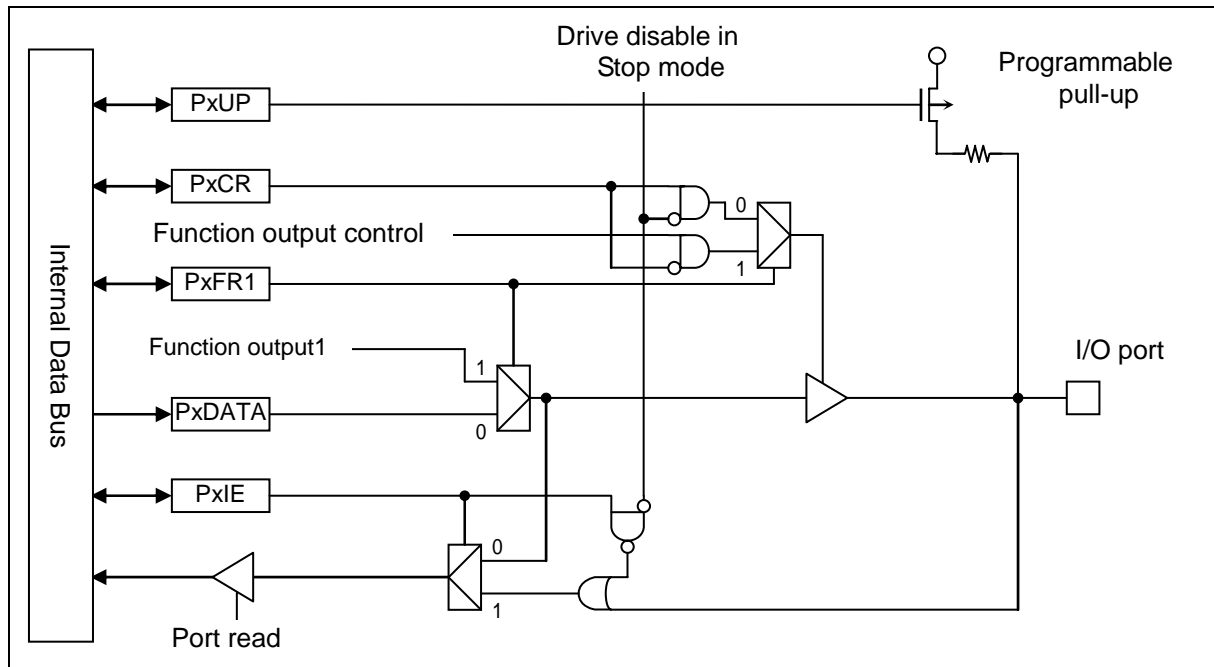


Fig 8.19 Type T19

8.3.21 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.

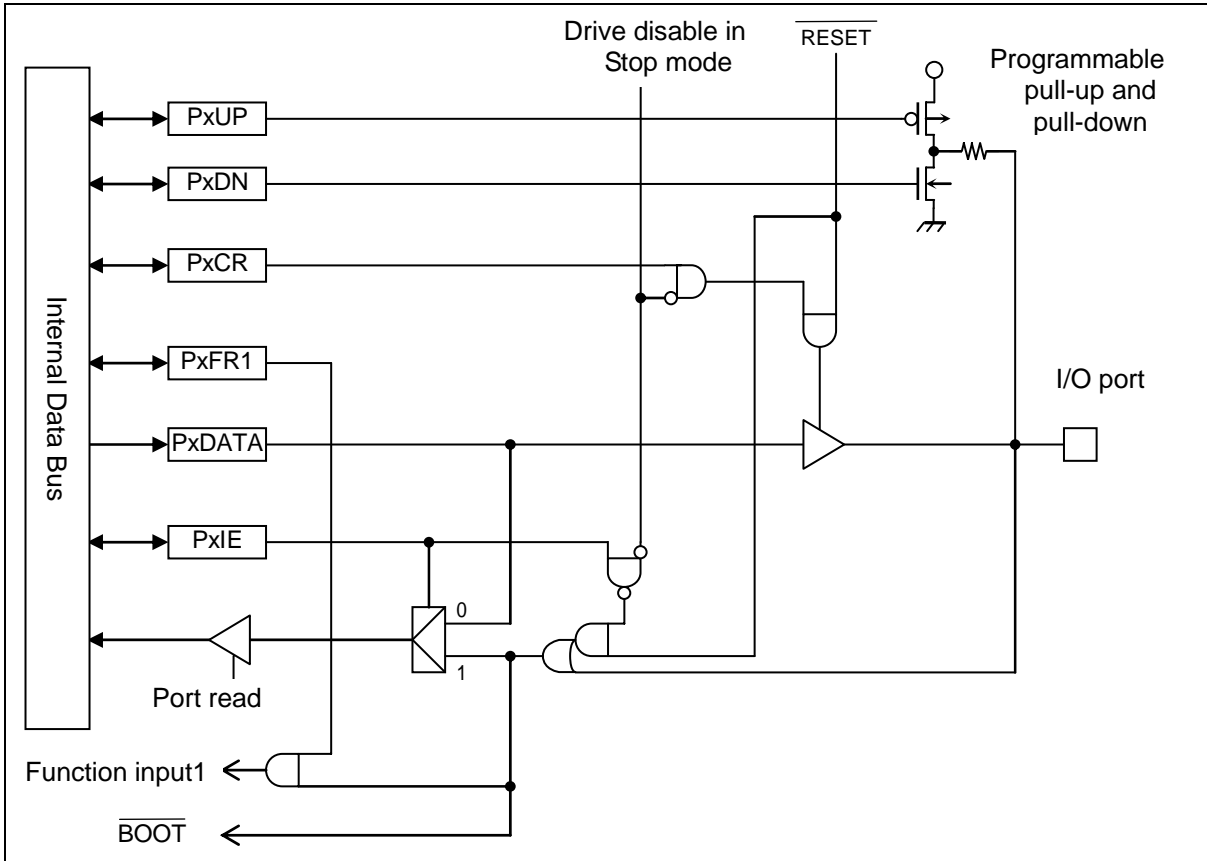


Fig 8.20 Type T20

## 9. 16-bit Timer/Event Counters (TMRBs)

### 9.1 Outline

TPM370 have 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)

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

## 9.2 Specification differences among channels

Channels (TMRB0 through TMRB7) operate independently and the functions are same except the differences as shown in Table 9-1 and Table 9-2. Therefore, the operational descriptions here are explained only for TMRB0.

Table 9-1 Differences in the Specifications of TMRB Modules (1)

| Specification<br>Channel | External pins                                |                            |
|--------------------------|--|----------------------------|
|                          | External clock/<br>capture trigger input pin | Timer flip-flop output pin |
| 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                    | TB5IN (shared with PD0)                      | TB5OUT (shared with PD1)   |
| TMRB6                    | TB6IN (shared with PA6)                      | TB6OUT (shared with PA5)   |
| TMRB7                    | TB7IN (shared with PF0)                      | TB7OUT (shared with PF1)   |

Table 9-2 Differences in the Specifications of TMRB Modules (2)

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

### 9.3 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.

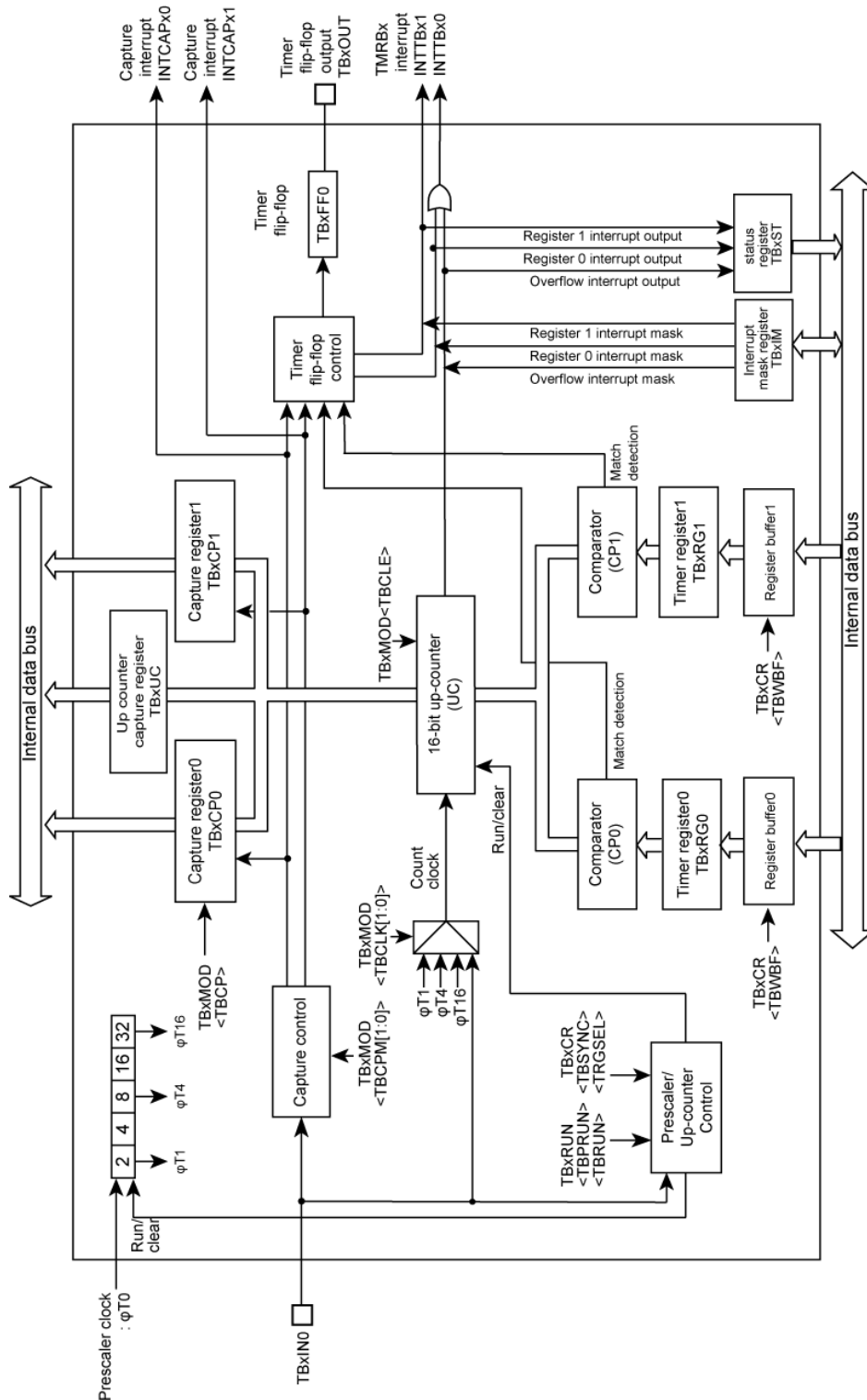


Fig. 9-1 TMRB0 Block Diagram (the same applies to channels 1 through 7)

## 9.4 Registers

### 9.4.1 TMRB registers

Table 9-3 shows the register names and addresses of each channel.

Table 9-3 TMRB registers

| Channel                       |                                  | TMRB0       |             | TMRB1       |             | TMRB2       |             | TMRB3       |             |
|-------------------------------|----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Specification                 |                                  |             |             |             |             |             |             |             |             |
| Register names<br>(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                       |                                  | TMRB4       |             | TMRB5       |             | TMRB6       |             | TMRB7       |             |
|-------------------------------|----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Specification                 |                                  |             |             |             |             |             |             |             |             |
| Register names<br>(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 | TB5FFCR     | 0x4001_0150 | 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 | TB5CP1      | 0x4001_016C | TB6CP1      | 0x4001_01AC | TB7CP1      | 0x4001_01EC |             |

9.4.1.1 TMRBn enable register (channels 0 through 7)

TMRBn enable register (n=0 to 7)

|                        |             |  |              |    |    |    |    |    |    |
|------------------------|-------------|--|--------------|----|----|----|----|----|----|
|                        |             | 31   | 30           | 29 | 28 | 27 | 26 | 25 | 24 |
| TbNEN<br>(0x4001_0xx0) | 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  |              |    |    |    |    |    |    |
|                        | Read/Write  | R/W  | R            |    |    |    |    |    |    |
|                        | After reset | 0  | 0            |    |    |    |    |    |    |
|                        | Function    | TMRBn operation<br>0: Disabled<br>1: Enabled | "0" is read. |    |    |    |    |    |    |

< 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.



9.4.1.2 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<br>0: Stop & clear<br>1: Count<br>* 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 counting 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.

## 9.4.1.3 TMRB control register (channels 0 through 7)

|                        |  | TMRBn control register (n=0 to 7) |            |              |                                 |              |  |  |    |
|------------------------|--|-----------------------------------|------------|--------------|---------------------------------|--------------|--|--|----|
|                        |  | 31                                | 30         | 29           | 28                              | 27           | 26   | 25   | 24 |
| TBnCR<br>(0x4001_0xx8) | 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<br>0: Disabled<br>1: Enabled | Write "0".                        | Write "0". | "0" is read. | IDLE<br>0: Stop<br>1: Operation | "0" is read. | External Trigger select<br>0: Rising edge<br>1: Falling edge | Counter Start select<br>0: Software start<br>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.

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

0: Select rising edge of external trigger.

1: Select falling edge of external trigger.

<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.

(Note1) TBnCR register must not be changed during Timer operation TBnRUN<TERUN>=1".

(Note2) When the external trigger start is used (<CSSELn>="1"), select <CSSELn> and <TRGSELn> before the setting of <TBnRUN>=<TBnPRUN>="1".

9.4.1.4 TMRB mode register (channels 0 thorough 7)

TMRBn mode register(n=0 to 7)

|                         |              |   |  |  |         |   |         |  |    |
|-------------------------|--------------|---|--|--|---------|---|---------|--|----|
| TbNMOD<br>(0x4001_0xxC) | 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)<br><br>0: Can be written separately<br>1: Must be written simultaneously | Software capture control<br><br>0: Software capture<br>1: Don't care | Capture timing<br>00: Disable<br>01: TBnIN ↑<br>10: TBnIN ↑ TBnIN ↓<br>11: Disable |         | Up-counter clear control<br>0: Clear/disable<br>1: clear/enable |         | Source clock select<br>00: TBnIN pin input<br>01: φT1<br>10: φT4<br>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)

< 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

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

- 00: software capture
- 01: Don't care

<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 register.

1: In case both registers 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".**

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

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

|                         |               |    |   |         |         |         |  |          |
|-------------------------|---------------|----|---|---------|---------|---------|--|----------|
| TnFFCR<br>(0x4001_0xx0) | 31            | 30 | 29  | 28      | 27      | 26      | 25   | 24       |
|                         | bit Symbol    |    |   |         |         |         |  |          |
|                         | Read/Write    | R  | R   | R       | R       | R       | R  | R        |
|                         | After reset   | 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        |
|                         | After reset   | 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  |          |
| After reset             | 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<br>0: Disable trigger<br>1: Enable trigger<br>When the up-counter value is taken into TBnCP1 |         |         |         | TBnFF0 control<br>00: Invert<br>01: Set<br>10: Clear<br>11: Don't care<br>* This is always read as "11." |          |

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

## 9.4.1.6 TMRB status register (channels 0 through 7)

TMRBn status register (n=0 to 7)

|                       |              |    |    |    |    |  |  |  |    |
|-----------------------|--------------|----|----|----|----|--|--|--|----|
| TnST<br>(0x4001_0xx4) | 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<br>1: Interrupt generated | 0: Interrupt not generated<br>1: Interrupt generated | 0: Interrupt not generated<br>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 TnST and the generation of interrupt enabled by TnIM is notified to the CPU. The flag is cleared by reading the TnST register.**

## 9.4.1.7 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<br>1: Interrupt is masked | 0: No Interrupt mask<br>1: Interrupt is masked | 0: No Interrupt mask<br>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.

9.4.1.8 TMRB read capture register (channels 0 through 7)

TBnUC0 read capture register (n=0 to 7)

|                         |             |  |       |       |       |       |       |      |      |
|-------------------------|-------------|--|-------|-------|-------|-------|-------|------|------|
| TBnUC0<br>(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.



9.4.1.9 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

9.4.1.10 TMRB capture register (channels 0 through 7)

TBnCP0capture register (n=0 to 7)

|                         |                                    |          |          |          |          |          |          |          |          |
|-------------------------|------------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| TBnCP0<br>(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 | TBnCP009 | TBnCP008 |
|                         | Read/Write                         | R        |          |          |          |          |          |          |          |
|                         | After reset                        | 0        |          |          |          |          |          |          |          |
| Function                | Timer capture value: 15-8 bit data |          |          |          |          |          |          |          |          |
|                         | 7                                  | 6        | 5        | 4        | 3        | 2        | 1        | 0        |          |
| bit Symbol              | TBnCP007                           | TBnCP006 | TBnCP005 | TBnCP004 | TBnCP003 | TBnCP002 | TBnCP001 | TBnCP000 |          |
| 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<br>(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 | TBnCP109 | TBnCP108 |
|                         | Read/Write                         | R        |          |          |          |          |          |          |          |
|                         | After reset                        | 0        |          |          |          |          |          |          |          |
| Function                | Timer capture value: 15-8 bit data |          |          |          |          |          |          |          |          |
|                         | 7                                  | 6        | 5        | 4        | 3        | 2        | 1        | 0        |          |
| bit Symbol              | TBnCP107                           | TBnCP106 | TBnCP105 | TBnCP104 | TBnCP103 | TBnCP102 | TBnCP101 | TBnCP100 |          |
| Read/Write              | R                                  |          |          |          |          |          |          |          |          |
| After reset             | 0                                  |          |          |          |          |          |          |          |          |
| Function                | Timer capture value: 7-0 bit data  |          |          |          |          |          |          |          |          |

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

## 9.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 9-1 and Table 9-2 . Therefore, the operational descriptions here are only for channel 0.

### 9.5.1 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter UC0. The prescaler input clock  $\phi_{T0}$  is  $f_{\text{periph}}$ ,  $f_{\text{periph}}/2$ ,  $f_{\text{periph}}/4$ ,  $f_{\text{periph}}/8$ ,  $f_{\text{periph}}/16$  or  $f_{\text{periph}}/32$  selected by SYSCR1<PRCLK2:0> in the CG. The peripheral clock,  $f_{\text{periph}}$ , is either  $f_{\text{gear}}$ , a clock selected by SYSCR1<FPSEL> in the CG, or  $f_c$ , which is a clock before it is divided by the clock gear.

The operation or the stop of a prescaler is set with TBORUN<TB0PRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 9-4 show prescaler output clock resolutions.

Table 9-4 Prescaler Output Clock Resolutions @fc = 80MHz

| Clear peripheral clock <FPSEL> | Clock gear value <GEAR2:0> | Prescaler clock selection <PRCK2:0> | Prescaler output clock resolution |                        |                         |
|--------------------------------|----------------------------|-------------------------------------|-----------------------------------|------------------------|-------------------------|
|                                |                            |                                     | $\phi T1$                         | $\phi T4$              | $\phi T16$              |
| 0 (fgear)                      | 000 (fc)                   | 000 (fper iph/1)                    | $fc/2^1(0.025\mu s)$              | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 100 (fper iph/16)                   | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 101 (fper iph/32)                   | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                | 100(fc/2)                  | 000 (fper iph/1)                    | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 100 (fper iph/16)                   | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 101 (fper iph/32)                   | $fc/2^7(1.6\mu s)$                | $fc/2^9(6.4\mu s)$     | $fc/2^{11}(25.6\mu s)$  |
|                                | 101(fc/4)                  | 000 (fper iph/1)                    | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 100 (fper iph/16)                   | $fc/2^7(1.6\mu s)$                | $fc/2^9(6.4\mu s)$     | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 101 (fper iph/32)                   | $fc/2^8(3.2\mu s)$                | $fc/2^{10}(12.8\mu s)$ | $fc/2^{12}(51.2\mu s)$  |
|                                | 110(fc/8)                  | 000 (fper iph/1)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 011 (fper iph/8)                    | $fc/2^7(1.6\mu s)$                | $fc/2^9(6.4\mu s)$     | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 100 (fper iph/16)                   | $fc/2^8(3.2\mu s)$                | $fc/2^{10}(12.8\mu s)$ | $fc/2^{12}(51.2\mu s)$  |
|                                |                            | 101 (fper iph/32)                   | $fc/2^9(6.4\mu s)$                | $fc/2^{11}(25.6\mu s)$ | $fc/2^{13}(102.4\mu s)$ |
| 1 (fc)                         | 000 (fc)                   | 000 (fper iph/1)                    | $fc/2^1(0.025\mu s)$              | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 100 (fper iph/16)                   | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 101 (fper iph/32)                   | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                | 100(fc/2)                  | 000 (fper iph/1)                    | -                                 | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 100 (fper iph/16)                   | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 101 (fper iph/32)                   | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                | 101(fc/4)                  | 000 (fper iph/1)                    | -                                 | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | -                                 | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 100 (fper iph/16)                   | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 101 (fper iph/32)                   | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |
|                                | 110(fc/8)                  | 000 (fper iph/1)                    | -                                 | -                      | $fc/2^5(0.4\mu s)$      |
|                                |                            | 001 (fper iph/2)                    | -                                 | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      |
|                                |                            | 010 (fper iph/4)                    | -                                 | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      |
|                                |                            | 011 (fper iph/8)                    | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      |
|                                |                            | 100 (fper iph/16)                   | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      |
|                                |                            | 101 (fper iph/32)                   | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  |

- (Note 1) The prescaler output clock  $\phi T_n$  must be selected as  $\phi T_n < f_{sys}$ . ( $\phi T_n$  is slower than  $f_{sys}$ ).
- (Note 2) Do not change the clock gear while the timer is operating.
- (Note 3) “—“ denotes a setting prohibited.

### 9.5.2 Up-counter (UC0)

UC0 is a 16-bit binary counter.

- Source clock

UC0 source clock, specified by TB0MOD<TB0CLK1:0>, can be selected from either three types  $\phi$ T1,  $\phi$ T4 and  $\phi$ T16 of prescaler output clock or the external clock of the TB0IN pin.

- Count start/ stop

Counter operation is specified by TB0RUN<TB0RUN>. UC0 starts counting if <TB0RUN> = "1", and stops counting and clears counter value if <TB0RUN> = "0".

- Timing to clear UC0

- 1) When a compare match is detected

By setting TB0MOD<TB0CLE> = "1", UC0 is cleared in case the comparator detects a match between counter value and the value set in TB0RG1. UC0 operates as a free-running counter if TB0MOD<TB0CLE> = "0".

- 2) When UC0 stops

UC0 stops counting and clears counter value if TB0RUN <TB0RUN> = "0".

- UC0 overflow

If UC0 overflow occurs, the INTTB00 overflow interrupt is generated.

### 9.5.3 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 UC0 up-counter, it outputs the match detection signal.

- Configuration

TB0RG0 and TB0RG1 of this timer registers are paired with register buffer - the double-buffered configuration. The two registers use TB0CR<TB0WBF> to control the enabling/disabling of double buffering. If <TB0WBF> = "0", double buffering is disabled and if <TB0WBF> = "1", it is enabled. If double buffering is enabled, data is transferred from register buffer to the TB0RG0 and TB0RG1 timer registers when there is a match between UC0 and TB0RG1.

- Default setting

The values of TB0RG0 and TB0RG1 become undefined after a reset. A reset disables the double buffer.

- Register setting

- 1) When not using double-buffering

To write data to the timer registers, either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits can be used.

- 2) When using double-buffering

TB0RG0/ TB0RG1 and the register buffers are assigned to the same address. If <TB0WBF> = "0," the same value is written to TB0RG0, TB0RG1 and each register buffer; if <TB0WBF> = "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 UC0 count value matching with TB0RG0 value.

INTTB01 is generated by UC0 count value matching with TB0RG1 value.

### 9.5.4 Capture Control

This is a circuit that controls the timing to latch UC0 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 UC0 up-counter into the capture register; specifically, UC0 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").

### 9.5.5 Capture Registers (TB0CP0, TB0CP1)

These are 16-bit registers for latching values from the UC0 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.

### 9.5.6 Up-counter capture register (TB0UC)

Other than the capturing functions shown above, the current count value of the UC0 can be captured by reading the TB0UC registers.

### 9.5.7 Comparators (CP0,CP1)

These are 16-bit comparators for detecting a match by comparing set values of the UC0 up-counter with set values of the TB0RG0 and TB0RG1 timer registers. If a match is detected, INTTB00 and INTTB01 are generated.

### 9.5.8 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.

### 9.5.9 Capture interrupt (INTCAP00, INTCAP01)

Interrupts INTCAP00 and INTCAP01 can be generated at the timing of latching values from the UC0 up-counter into the TB0CP0 and TB0CP1 capture registers. The interrupt setting is specified by the CPU.



## 9.6 Description of Operations for Each Mode

### 9.6.1 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 register,

### 9.6.2 16-bit Event Counter Mode

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

### 9.6.3 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 (UC0) 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:

$$(\text{Set value of TB0RG0}) < (\text{Set value of TB0RG1})$$

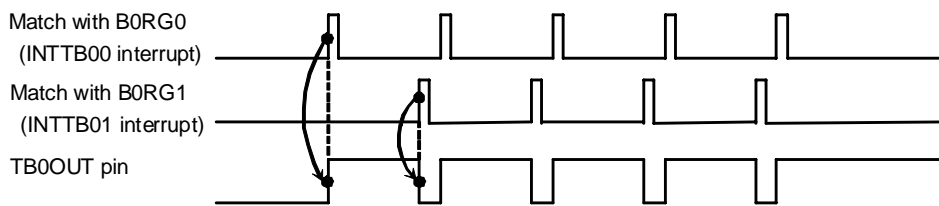


Fig. 9-2 Example of Output of Programmable Square Wave (PPG)

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.

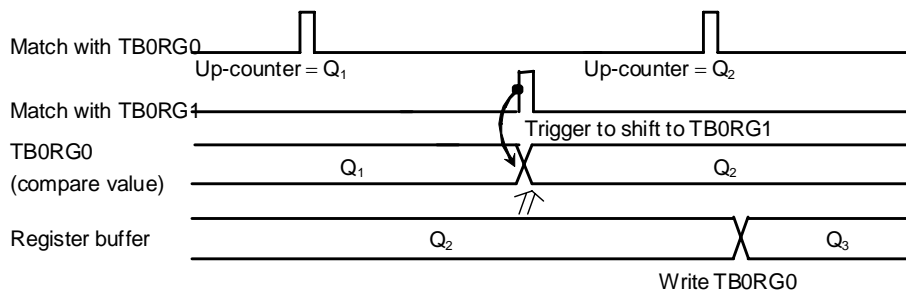


Fig. 9-3 Register Buffer Operation

The block diagram of this mode is shown below.

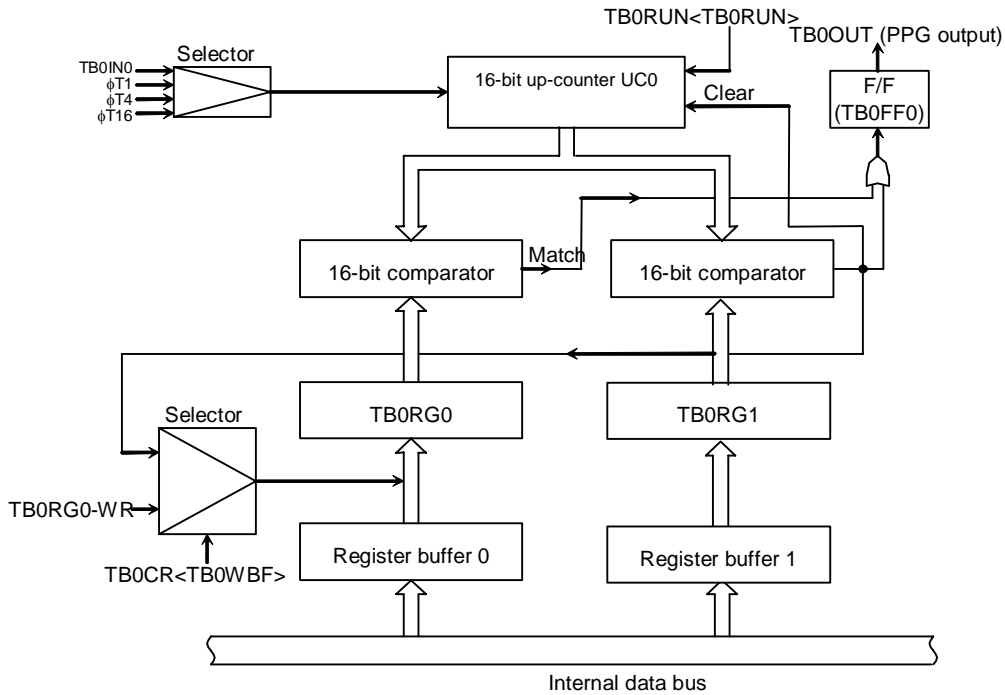


Fig. 9-4 Block Diagram of 16-bit PPG Mode

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

|         | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|---------|-----|---|---|---|---|---|---|---|---|
| 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   |
| TBORG0  | ← * | * | * | * | * | * | * | * | Specifies a duty. (16 bits *32-bits register length)  |
| TBORG1  | ← * | * | * | * | * | * | * | * | Specifies a cycle. (16 bits *32-bits register length)   |
| TB0CR   | ← 1 | 0 | X | 0 | 0 | 0 | 0 | 0 | Enables the TBORG0 double buffering.<br>(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 TBORG0 or TBORG1 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.  |
| PACR    | ← - | - | - | - | - | - | 1 | - | } Assigns PA0 to output and TB0OUT  |
| PAFR1   | ← - | - | - | - | - | - | 1 | - |   |
| TB0RUN  | ← * | * | * | * | * | 1 | X | 1 |   |

X; Don't care -; no change

### 9.6.4 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 (UC0) 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 (UC0) 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.

Figure 9-5 shows one-shot pulse generation, with annotations showing (d) and (p).

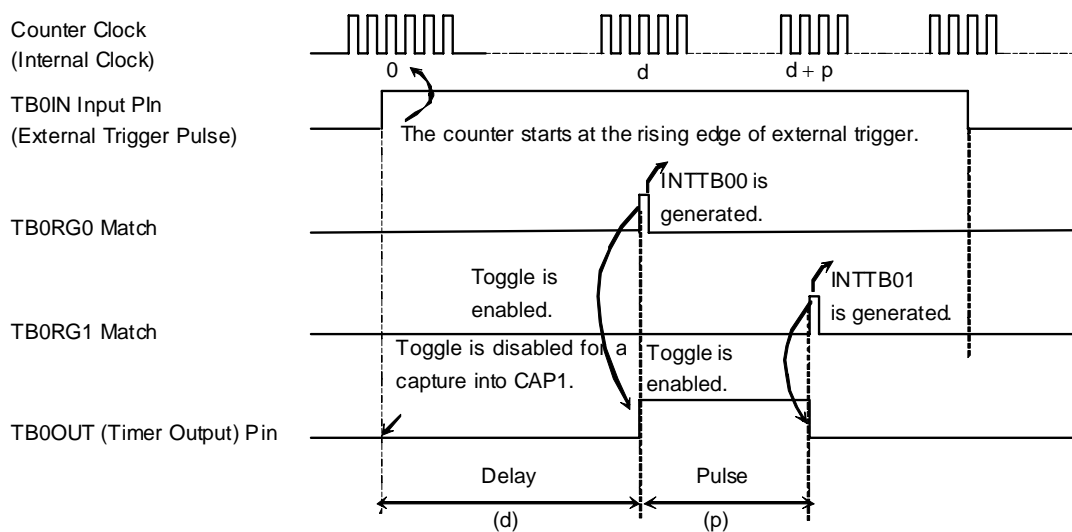


Fig. 9-5 One-shot pulse generation using an external count start trigger (with a delay)

### 9.7 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

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<TBE1T1, TBE0T1>) must be set to “11.” This enables triggering the timer flip-flop (TB0FF0) to reverse when UC0 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. 9-6 One-shot Pulse Output (With Delay).”

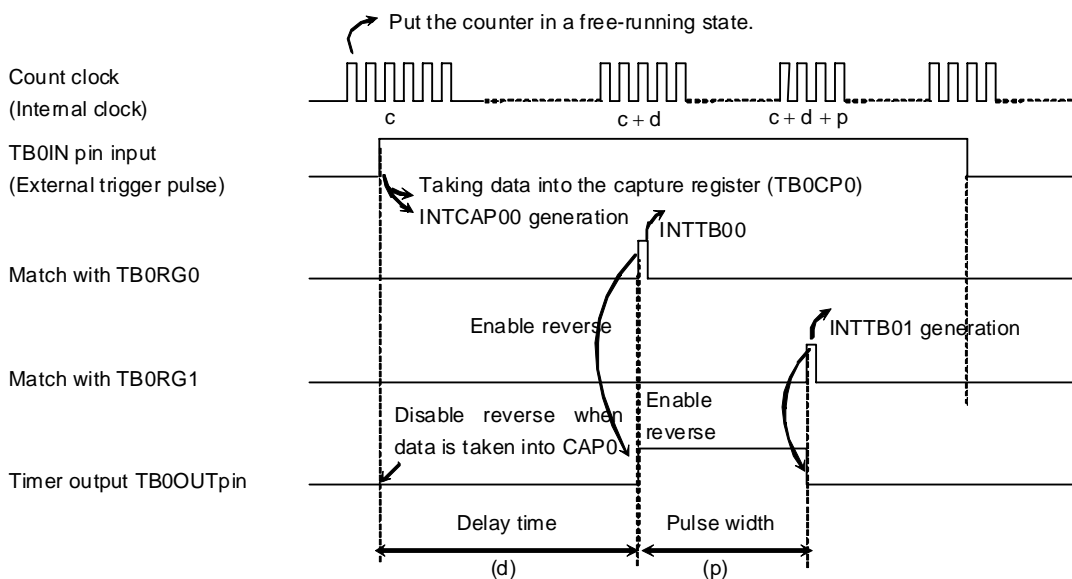


Fig. 9-6 One-shot Pulse Output (With Delay)

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 UC0 matches with TB0RG1, and is disabled by generating the INTTB01 interrupt.

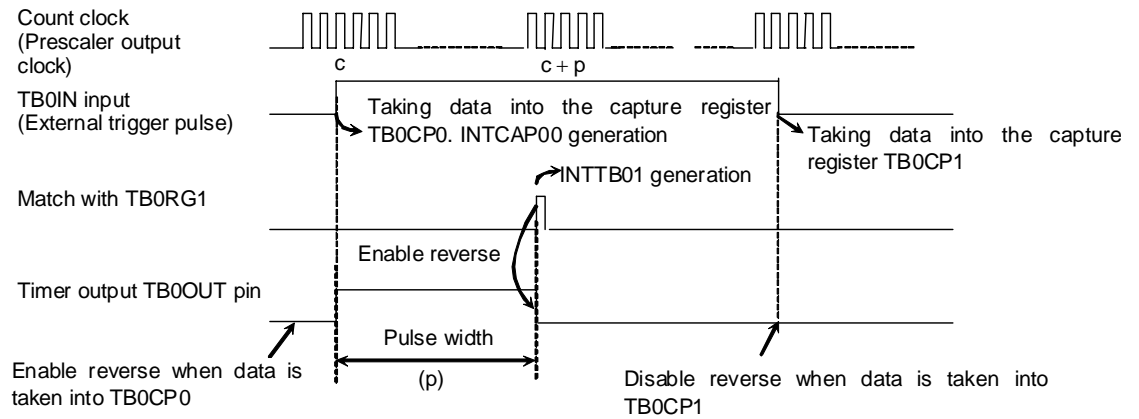


Fig. 9-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

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 (UC0) 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  $\mu$ s, the pulse width is  $100 \times 0.5 \mu\text{s} = 50 \mu\text{s}$ .

Caution must be exercised when measuring pulse widths exceeding the UC0 maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.

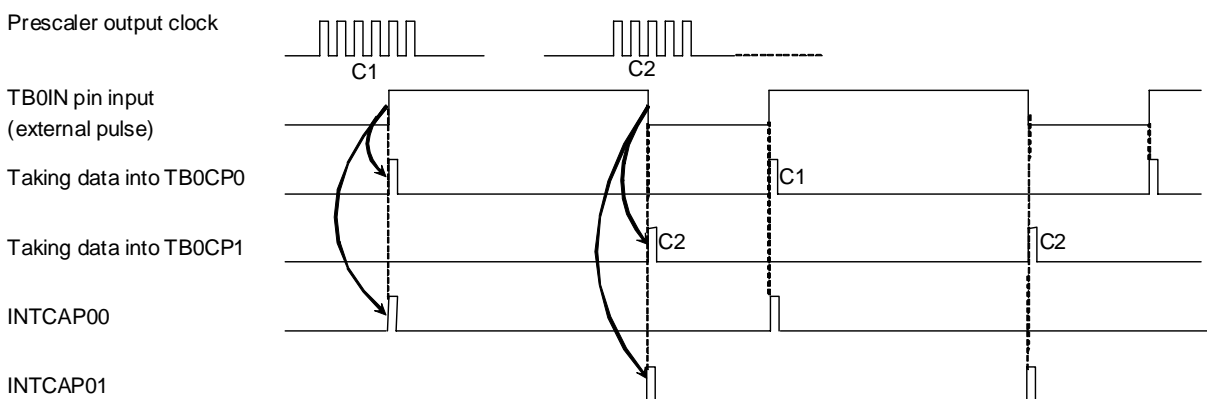


Fig. 9-8 Pulse Width Measurement

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. 9-8 Pulse Width Measurement” and this difference is multiplied by the cycle of the prescaler output clock to obtain the “Low” level width.

## 10 Serial Channel (SIO)

### 10.1 Features

This device has four serial I/O channels: SIO0 to SIO3. Each channel operates in either the UART mode (asynchronous communication) or the I/O interface mode (synchronous communication) which is selected by the user.

I/O interface mode — Mode 0: This is the mode to transmit and receive I/O data and associated synchronization signals (SCLK) to extend I/O.

Asynchronous (UART) mode: — Mode 1: TX/RX Data Length: 7 bits  
— Mode 2: TX/RX Data Length: 8 bits  
— Mode 3: TX/RX Data Length: 9 bits

In the above modes 1 and 2, parity bits can be added. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). Fig. 10-2 shows the block diagram of SIO0.

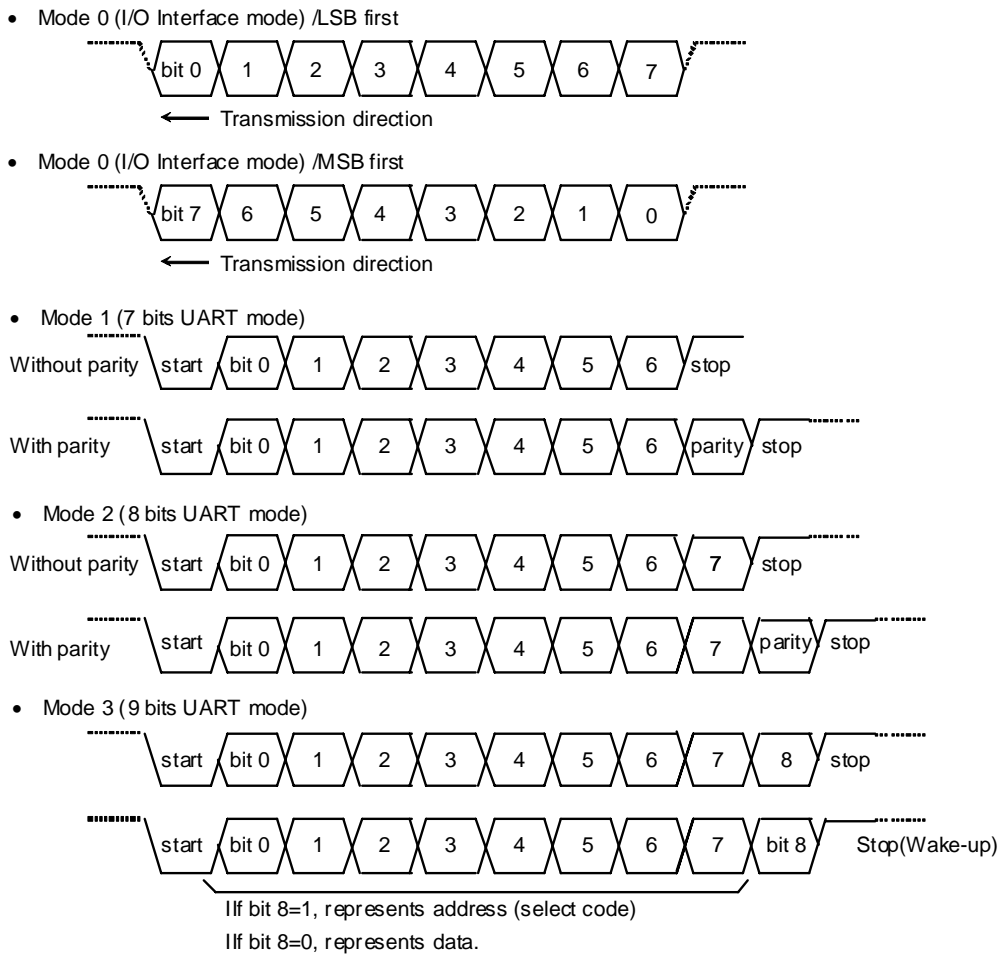
Each channel consists of a prescaler, a serial clock generation circuit, a receive buffer, its control circuit, a transmit buffer and its control circuit. Each channel functions independently.

As the SIOs 0 to 3 operate in the same way, only SIO0 is described here.



Table 10-1 Difference in the Specifications of SIO Modules

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



**Fig. 10-1 Data Format**

10.2 Block Diagram (Channel 0)

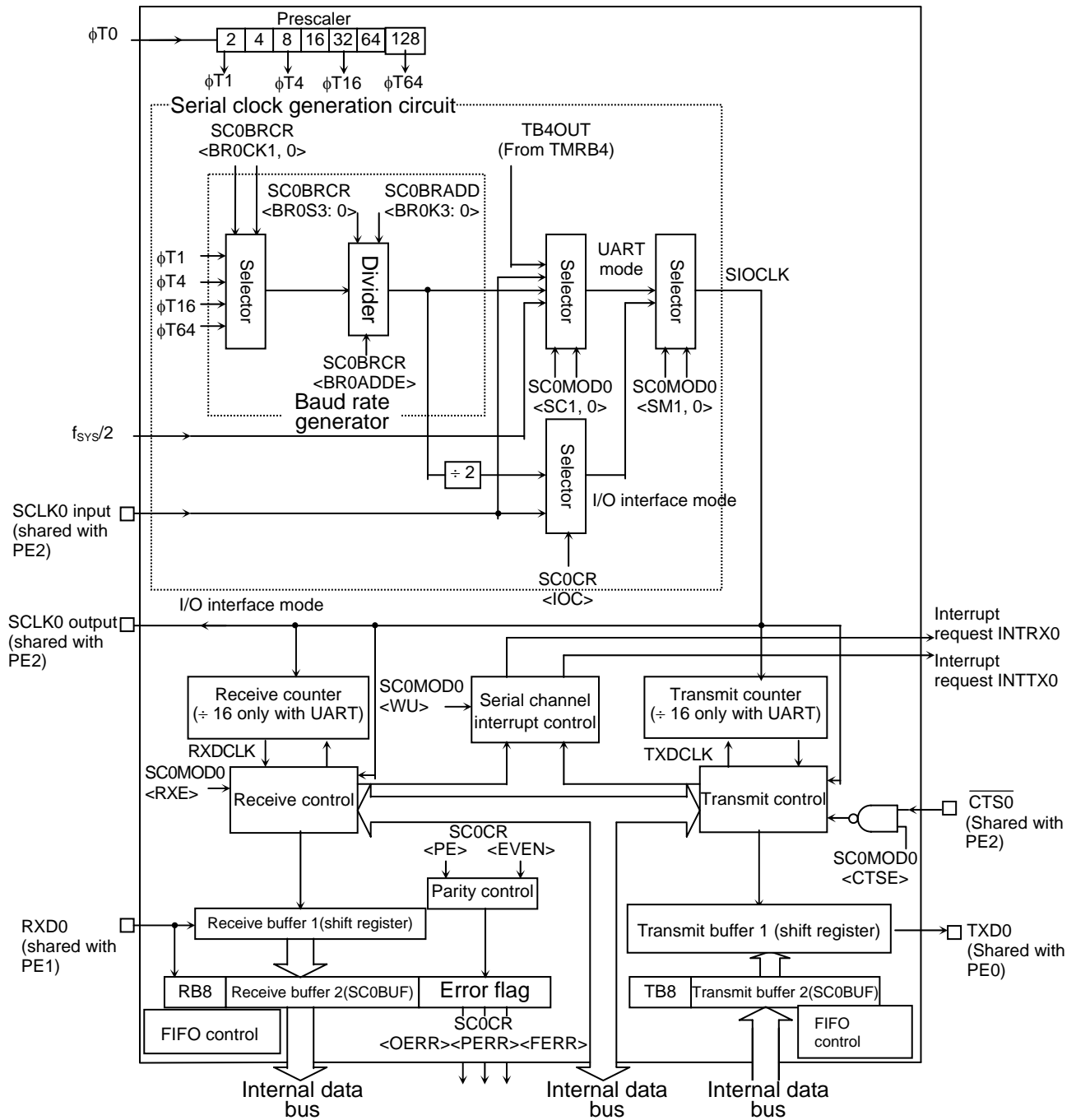


Fig. 10-2 SIO0 Block Diagram

## 10.3 Operation of Each Circuit (Channel 0)

### 10.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock  $\phi T0$  to the prescaler is selected by SYSCR1 of CG <PRCK2:0> to provide the frequency of either  $f_{\text{periph}}/1$ ,  $f_{\text{periph}}/2$ ,  $f_{\text{periph}}/4$ ,  $f_{\text{periph}}/8$ ,  $f_{\text{periph}}/16$  or  $f_{\text{periph}}/32$ .

The clock frequency  $f_{\text{periph}}$  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 10-2 list the prescaler output clock resolution.

**Table 10-2 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 |                        |                         |                         |
|--------------------------------|----------------------------|-------------------------------------|-----------------------------------|------------------------|-------------------------|-------------------------|
|                                |                            |                                     | $\phi T1$                         | $\phi T4$              | $\phi T16$              | $\phi T64$              |
| 0 (fgear)                      | 000 (fc)                   | 000 (fperiph/1)                     | $fc/2^1(0.025\mu s)$              | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      | $fc/2^7(1.6\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      | $fc/2^8(3.2\mu s)$      |
|                                |                            | 010 (fperiph/4)                     | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 011 (fperiph/8)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 101 (fperiph/32)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                | 100 (fc/2)                 | 000 (fperiph/1)                     | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      | $fc/2^8(3.2\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 010 (fperiph/4)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 011 (fperiph/8)                     | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                |                            | 101 (fperiph/32)                    | $fc/2^7(1.6\mu s)$                | $fc/2^9(6.4\mu s)$     | $fc/2^{11}(25.6\mu s)$  | $fc/2^{13}(102.4\mu s)$ |
|                                | 101 (fc/4)                 | 000 (fperiph/1)                     | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 010 (fperiph/4)                     | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 011 (fperiph/8)                     | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^7(1.6\mu s)$                | $fc/2^9(6.4\mu s)$     | $fc/2^{11}(25.6\mu s)$  | $fc/2^{13}(102.4\mu s)$ |
|                                |                            | 101 (fperiph/32)                    | $fc/2^8(3.2\mu s)$                | $fc/2^{10}(12.8\mu s)$ | $fc/2^{12}(51.2\mu s)$  | $fc/2^{14}(204.8\mu s)$ |
|                                | 110 (fc/8)                 | 000 (fperiph/1)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 001 (fperiph/2)                     | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 010 (fperiph/4)                     | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                |                            | 011 (fperiph/8)                     | $fc/2^7(1.6\mu s)$                | $fc/2^9(6.4\mu s)$     | $fc/2^{11}(25.6\mu s)$  | $fc/2^{13}(102.4\mu s)$ |
|                                |                            | 100 (fperiph/16)                    | $fc/2^8(3.2\mu s)$                | $fc/2^{10}(12.8\mu s)$ | $fc/2^{12}(51.2\mu s)$  | $fc/2^{14}(204.8\mu s)$ |
|                                |                            | 101 (fperiph/32)                    | $fc/2^9(6.4\mu s)$                | $fc/2^{11}(25.6\mu s)$ | $fc/2^{13}(102.4\mu s)$ | $fc/2^{15}(409.6\mu s)$ |
| 1 (fc)                         | 000 (fc)                   | 000 (fperiph/1)                     | $fc/2^1(0.025\mu s)$              | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      | $fc/2^7(1.6\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      | $fc/2^8(3.2\mu s)$      |
|                                |                            | 010 (fperiph/4)                     | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 011 (fperiph/8)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 101 (fperiph/32)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                | 100 (fc/2)                 | 000 (fperiph/1)                     | -                                 | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      | $fc/2^7(1.6\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | $fc/2^2(0.05\mu s)$               | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      | $fc/2^8(3.2\mu s)$      |
|                                |                            | 010 (fperiph/4)                     | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 011 (fperiph/8)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 101 (fperiph/32)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                | 101 (fc/4)                 | 000 (fperiph/1)                     | -                                 | $fc/2^3(0.1\mu s)$     | $fc/2^5(0.4\mu s)$      | $fc/2^7(1.6\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | -                                 | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      | $fc/2^8(3.2\mu s)$      |
|                                |                            | 010 (fperiph/4)                     | $fc/2^3(0.1\mu s)$                | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 011 (fperiph/8)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 101 (fperiph/32)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |
|                                | 110 (fc/8)                 | 000 (fperiph/1)                     | -                                 | -                      | $fc/2^5(0.4\mu s)$      | $fc/2^7(1.6\mu s)$      |
|                                |                            | 001 (fperiph/2)                     | -                                 | $fc/2^4(0.2\mu s)$     | $fc/2^6(0.8\mu s)$      | $fc/2^8(3.2\mu s)$      |
|                                |                            | 010 (fperiph/4)                     | -                                 | $fc/2^5(0.4\mu s)$     | $fc/2^7(1.6\mu s)$      | $fc/2^9(6.4\mu s)$      |
|                                |                            | 011 (fperiph/8)                     | $fc/2^4(0.2\mu s)$                | $fc/2^6(0.8\mu s)$     | $fc/2^8(3.2\mu s)$      | $fc/2^{10}(12.8\mu s)$  |
|                                |                            | 100 (fperiph/16)                    | $fc/2^5(0.4\mu s)$                | $fc/2^7(1.6\mu s)$     | $fc/2^9(6.4\mu s)$      | $fc/2^{11}(25.6\mu s)$  |
|                                |                            | 101 (fperiph/32)                    | $fc/2^6(0.8\mu s)$                | $fc/2^8(3.2\mu s)$     | $fc/2^{10}(12.8\mu s)$  | $fc/2^{12}(51.2\mu s)$  |

**(Note 1) The prescaler output clock  $\phi T_n$  must be selected so that the relationship “ $\phi T_n < f_{sys}$ ” is satisfied (so that  $\phi T_n$  is slower than  $f_{sys}$ ).**

**(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.,  $\phi T_1$ ,  $\phi T_4$ ,  $\phi T_{16}$  and  $\phi T_{64}$ , supplied from the prescaler output clock.

### 10.3.2 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  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$  or  $\phi T64$  clock supplied from the 7-bit prescaler. This input clock selection is made by setting the baud rate generator control register, SC0BRCCR <BR0CK1:0>.

The baud rate generator contains built-in dividers for divide by 1,  $N + m/16$  ( $N=2\sim 15$ ,  $m=1\sim 15$ ), and 16. The division is performed according to the settings of the baud rate generator control registers SC0BRCCR <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 SC0BRCCR <BR0ADDE> = 0,

The setting of SC0BRADD <BR0K3:0> is ignored and the counter is divided by N where N is the value set to SC0BRCCR <BR0S3:0>. ( $N = 1$  to 16).

- 2) If SC0BRCCR <BR0ADDE> = 1,

The  $N + (16 - K)/16$  division function is enabled and the division is made by using the values N (set in SC0BRCCR <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 SC0BRCCR<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 SC0BRCCR <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 1.25 Mbps.

The  $f_{\text{sys}}$  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 2.5 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 10Mbps. (If double buffering is not used, the highest baud rate will be 5.0 Mbps).

- Example baud rate setting:

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

Selecting  $f_c = 78.642$  MHz for  $f_{periph}$ , setting  $\phi T0$  to  $f_{periph}/16$ , using the baud rate generator input clock  $\phi T1$ , 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 ( $f_c$ )  
High speed clock gear : x 1 ( $f_c$ )  
Prescaler clock :  $f_{periph}/16$  ( $f_{periph} = f_{sys}$ )

$$\begin{aligned} \text{Baud rate} &= \frac{f_c/32}{4} / 16 \\ &= 78.642 \times 10^6 \div 32 \div 4 \div 16 = 38400 \text{ (bps)} \end{aligned}$$

**(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  $f_c = 9.6$  MHz for  $f_{periph}$ , setting  $\phi T0$  to  $f_{periph}/8$ , using the baud rate generator input clock  $\phi T1$ , 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  $\left\{ \begin{array}{l} \text{System clock} : \text{High-speed } (f_c) \\ \text{High-speed clock gear} : \quad \times 1 (f_c) \\ \text{Prescaler clock} : f_{periph}/4 (f_{periph} = f_{sys}) \end{array} \right.$

$$\begin{aligned} \text{Baud rate} &= \frac{f_c/16}{7 + \frac{(16-3)}{16}} / 16 \\ &= 9.6 \times 10^6 \div 16 \div (7 + \frac{13}{16}) \div 16 = 4800 \text{ (bps)} \end{aligned}$$

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  $2/f_{sys}$ .

Moreover, please set to become it to the highest baud rate less than 1.25 (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 >  $6/f_{sys}$

Moreover, it is necessary to adjust the highest baud rate to less than 6.66 (Mbps).

When double buffering is not used, it is necessary to satisfy the following relationship:

External clock input period >  $8/f_{sys}$

Moreover, it is necessary to adjust the highest baud rate to less than 5.0 (Mbps).

The baud rate examples for the UART mode are shown in Table 10-3 and Table 10-4.

**Table 10-3 Selection of UART Baud Rate**

(Using the baud rate generator with SC0BRCR &lt;BR0ADDE&gt; = 0)

Unit: (kbps)

| fc [MHz] | Input clock                                    |                     |                      |                       |                        |
|----------|--|---------------------|----------------------|-----------------------|------------------------|
|          | Divide ratio N<br>(Set to SC0BRCR <BR0S3 : 0>) | $\phi T1$<br>(fc/4) | $\phi T4$<br>(fc/16) | $\phi T16$<br>(fc/64) | $\phi T64$<br>(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  $f_{\text{periph}}/2$ .

**Table 10-4 Selection of UART Baud Rate**(The TMRB4 timer output (internal TB4OUT) is used with the timer input clock set to  $\phi T1$ .)

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:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by SYSCR0<PRCK2:0>}}{\text{TB4RG} \times 2 \times 2 \times 16}$$

↑ (When input clock to the timer TMRB4 is  $\phi T1$ )

**(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  $f_{\text{periph}}/2$ .

### 10.3.3 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.

### 10.3.4 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.

### 10.3.5 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 rising 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.

### 10.3.6 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 “RBFL”) 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 (SC0FCNF <CNFG> = 0 and SC0MOD1<FDPX1:0> = 01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SC0FCNF <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 <RBFL> 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 (SC0BUF).

By disabling the second receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (SC0FCNF <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 SC0FNCNF <RFST>) and both the first and second receive buffers contain valid data. Also in this case, if SC0FCNF <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.”

### 10.3.7 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.

### 10.3.8 Receive FIFO Operation

- ① I/O interface mode with SCLK output:

The following example describes the case a 4-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, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing “1” to the RXE bit. After receiving 4 bytes, the RXE bit is automatically cleared and the receive operation is stopped (SCLK is stopped).

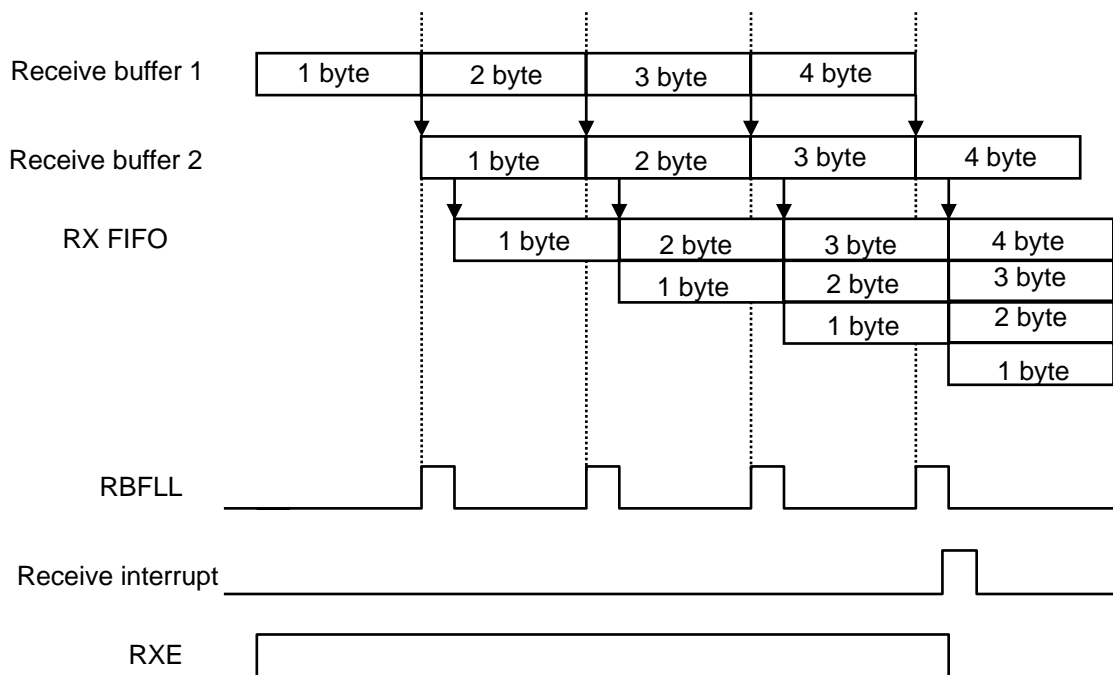


Fig. 10-3 Receive FIFO Operation

② I/O interface mode with SCLK input:

The following example describes the case a 4-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, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. After receiving 4 bytes, 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.

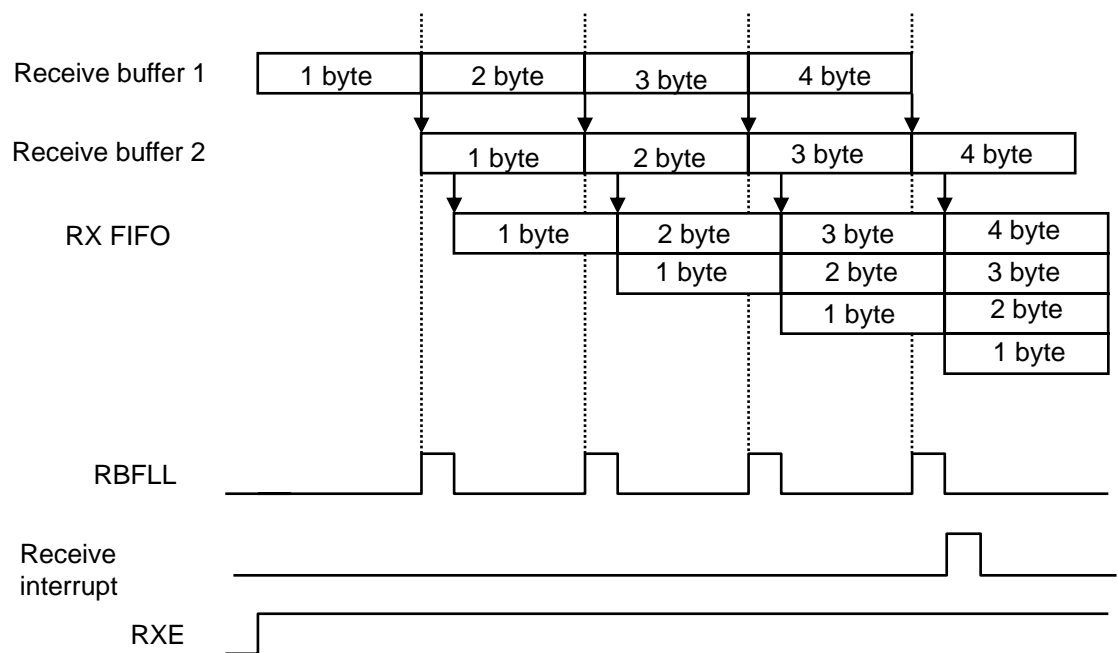


Fig. 10-4 Receive FIFO Operation

### 10.3.9 Transmit Counter

The transmit counter is a 4-bit binary counter used in the asynchronous communication (UART) mode. It is counted by SIOCLK as in the case of the receive counter and generates a transmit clock (TXDCLK) on every 16th clock pulse.

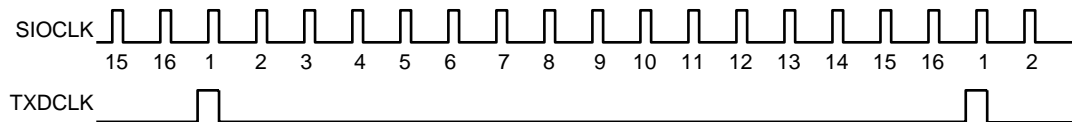


Fig. 10-5 Transmit Clock Generation

### 10.3.10 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 rising 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:

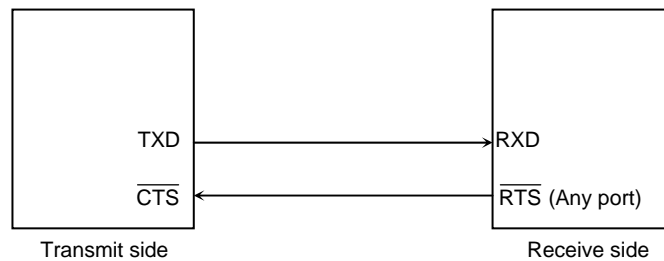
When the CPU writes data to the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock (TXDSFT) is also generated.

- Handshake function

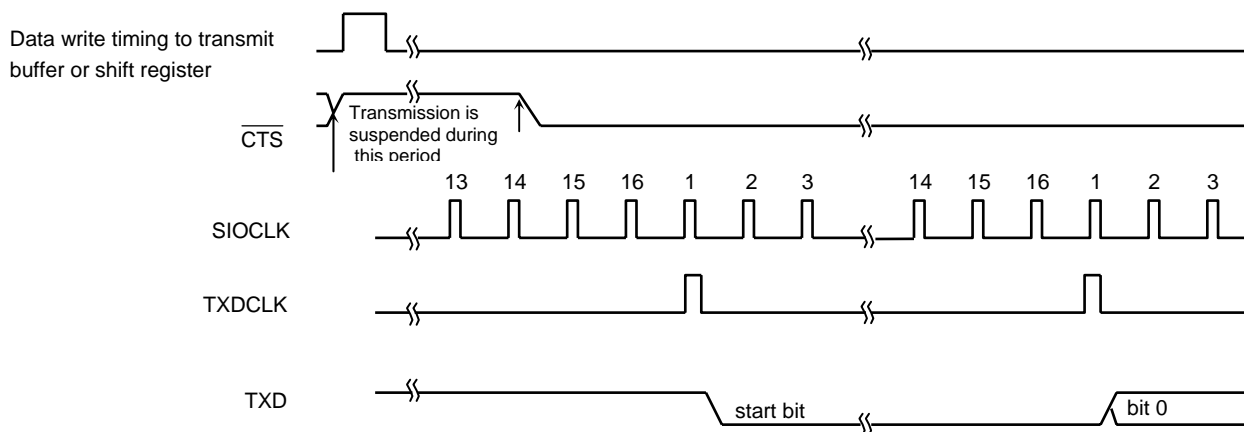
The  $\overline{\text{CTS}}$  pin enables frame by frame data transmission so that overrun errors can be prevented. This function can be enabled or disabled by SC0MOD0 <CTSE>.

When the  $\overline{\text{CTS0}}$  pin is set to the “H” level, the current data transmission can be completed but the next data transmission is suspended until the  $\overline{\text{CTS0}}$  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  $\overline{\text{RTS}}$  pin is provided, a handshake control function can be easily implemented by assigning a port for the  $\overline{\text{RTS}}$  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. 10-6 Handshake Function**



**Fig. 10-7 CTS (Clear to Transmit) Signal Timing**

**(Note 1)** If the  $\overline{\text{CTS}}$  signal is set to “H” during transmission, the next data transmission is suspended after the current transmission is completed.

**(Note 2)** Data transmission starts on the first falling edge of the TXDCLK clock after  $\overline{\text{CTS}}$  is set to “L.”



### 10.3.11 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 (SC0BUF) is moved to Transmit Buffer 1 (shift register).

If the transmit FIFO has been disabled (SC0FCNF <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.

### 10.3.12 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.

### 10.3.13 Transmit FIFO Operation

- ① I/O interface mode with SCLK output (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

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.

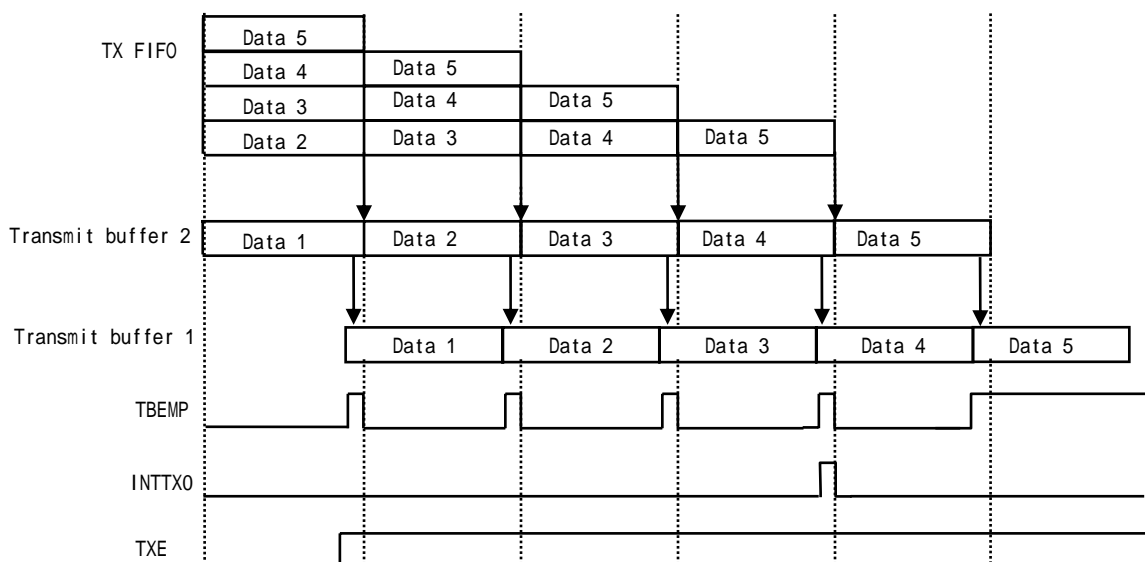


Fig. 10-8 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

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.

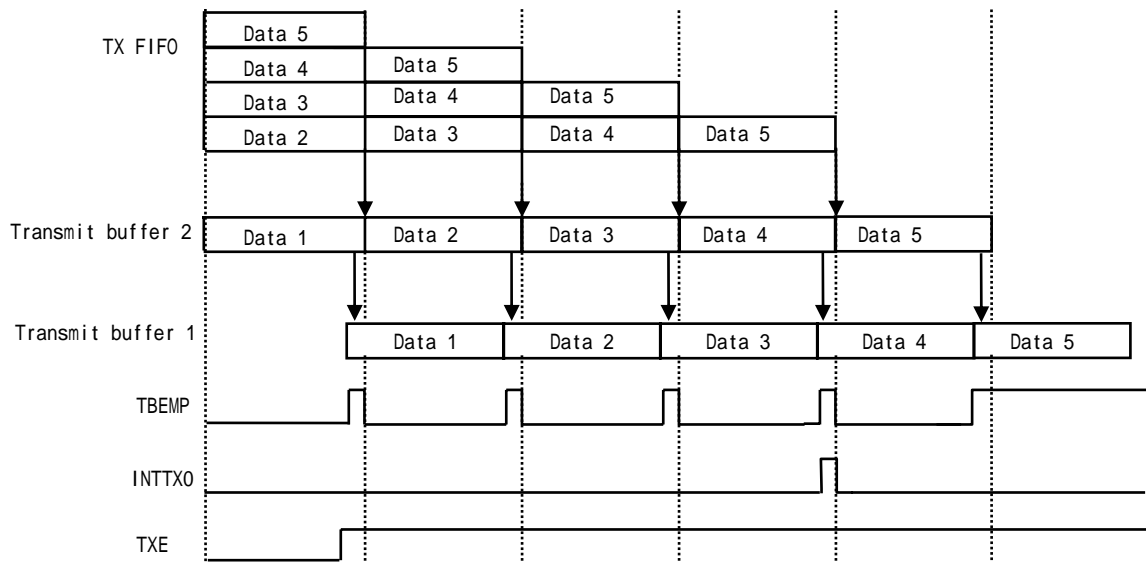


Fig. 10-9 Transmit FIFO Operation

### 10.3.14 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.

### 10.3.15 Error Flag

Three error flags are provided to improve 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.

|   |
|---|
| <b>Note)</b> 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 <SBLLEN> (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<br>(SCLK input)  | OERR       | Overrun error flag             |
|                                | PERR       | Underrun error flag (WBUF = 1) |
|                                |            | Fixed to 0 (WBUF = 0)          |
| FERR                           | Fixed to 0 |                                |
| I/O Interface<br>(SCLK output) | OERR       | Operation undefined            |
|                                | PERR       | Operation undefined            |
|                                | FERR       | Fixed to 0                     |

### 10.3.16 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.

### 10.3.17 Stop Bit Length

In the UART transmission mode, the stop bit length can be set to either 1 or 2 bits by bit 4 <SBLLEN> of the SC0MOD2 register. In the receive side, the stop bit is always regarded as 1 bit.

### 10.3.18 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.

### 10.3.19 Configurations of Transmit/Receive Buffer

|                                |                 | <WBUF> = 0 | <WBUF> = 1 |
|--------------------------------|-----------------|------------|------------|
| UART                           | Transmit buffer | Single     | Double     |
|                                | Receive buffer  | Double     | Double     |
| I/O Interface<br>(SCLK input)  | Transmit buffer | Single     | Double     |
|                                | Receive buffer  | Double     | Double     |
| I/O Interface<br>(SCLK output) | Transmit buffer | Single     | Double     |
|                                | Receive buffer  | Single     | Double     |

### 10.3.20 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>,<RBFL>,<TXRUN> of mode registers and SC0CR<OERR>, <PERR>, <FERR> of control registers and internal circuit is initialized. Other states are maintained.

10.3.21 Signal Generation Timing

① UART Mode:

Receive Side

| Mode                            | 9-bit                                 | 8-bit with parity                          | 8-bit, 7-bit, and 7-bit with parity        |
|---------------------------------|---------------------------------------|--|--|
| Interrupt generation timing     | Around the center of the 1st stop bit | Around the center of the 1st stop bit      | Around the center of the 1st stop bit      |
| Framing error generation timing | Around the center of the stop bit     | Around the center of the stop bit          | Around the center of the stop bit          |
| Parity error generation timing  | —                                     | Around the center of the last (parity) bit | Around the center of the last (parity) bit |
| Overrun error generation timing | Around the center of the stop bit     | Around the center of the stop bit          | Around the center of the stop bit          |

Transmit Side

| Mode                                     | 9-bit   | 8-bit with parity  | 8-bit, 7-bit, and 7-bit with parity   |
|--|---|--|---|
| Interrupt generation timing (<WBUF> = 0) | Just before the stop bit is sent  | Just before the stop bit is sent   | Just before the stop bit is sent  |
| Interrupt generation timing (<WBUF> = 1) | Immediately after data is moved to transmit buffer 1 (just before start bit transmission) | Immediately after data is moved to transmit buffer 1 (just before start bit transmission). | Immediately after data is moved to transmit buffer 1 (just before start bit transmission) |

I/O interface mode:

Receive Side

|  |                  |   |
|--|------------------|---|
| Interrupt generation timing (<WBUF> = 0) | SCLK output mode | Immediately after the rising edge of the last SCLK  |
|  | SCLK input mode  | Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively).                            |
| Interrupt generation timing (<WBUF> = 1) | SCLK output mode | Immediately after the rising edge of the last SCLK (just after data transfer to receive buffer 2) or just after receive buffer 2 is read. |
|  | SCLK input mode  | Immediately after the rising edge or falling edge of the last SCLK (right after data is moved to receive buffer 2).                       |
| Overrun error generation timing          | SCLK input mode  | Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)                             |

Transmit Side

|  |                  |  |
|--|------------------|--|
| Interrupt generation timing (<WBUF> = 0) | SCLK output mode | Immediately after the rising edge of the last SCLK   |
|  | SCLK input mode  | Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)  |
| Interrupt generation timing (<WBUF> = 1) | SCLK output mode | Immediately after the rising edge of the last SCLK or just after data is moved to Transmit Buffer 1            |
|  | SCLK input mode  | Immediately after the rising or falling edge of the last SCLK or just after data is moved to Transmit Buffer 1 |
| Under-run error generation timing        | SCLK input mode  | Immediately after the falling or rising edge of the next SCLK  |

**(Note 1) Do not modify any control register when data is being sent or received (in a state ready to transmit or receive).**

**(Note 2) Do not stop the receive operation (by setting SC0MOD0 <RXE> = "0") when data is being received.**

**(Note 3) Do not stop the transmit operation (by setting SC0MOD1 <TXE> = "0") when data is being transmitted.**

## 10.4 Register Description (Only for Channel 0)

The channel 0 registers are described here. Each register for all the channels operates in the same way.

### 10.4.1 Enable register

|       | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
|-------|-------------|---|---|---|---|---|---|---|--|
| SC0EN | bit Symbol  |   |   |   |   |   |   |   | SIOE                                     |
|       | Read/Write  |   |   |   |   |   |   |   | R/W                                      |
|       | After reset |   |   |   |   |   |   |   | 0  |
|       | Function    |   |   |   |   |   |   |   | SIO operation<br>0:disabled<br>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.

### 10.4.2 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       |   |
|--------|------------|---------|---------|---------|---------|---------|---------|---------|---|
| SC0BUF | TB7/RB7    | TB6/RB6 | TB5/RB5 | TB4/RB4 | TB3/RB3 | TB2/RB2 | TB1/RB1 | TB0/RB0 |   |
|        | Read/Write |         |         |         |         |         |         |         | R/W   |
|        | 0          | 0       | 0       | 0       | 0       | 0       | 0       | 0       |   |
|        | Function   |         |         |         |         |         |         |         | TB7~0 : Transmit buffer/FIFO<br>RB7~0 : Receive buffer/FIFO |

<TB7:0> Transmit buffer (at WR operation).

<RB7:0> Receive buffer (at RD operation).



## 10.4.3 Control register

|             | 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)<br>0: Odd<br>1: Even | Add parity (For UART)<br>0: Disabled<br>1: Enabled | 0: Normal operation<br>1: Error |                 |         | 0: SCLK0<br><br>1: SCLK0<br> | (For I/O interface)<br>0: Baud rate generator<br>1: SCLK0 pin input |
|             |                               |  |  | Overrun                         | Parity/underrun | Framing |  |   |

<RB8>: 9<sup>th</sup> 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.**

10.4.4 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                 |   |  |   |  |     |  |     |
| SC0MOD0 | After reset | 0                   | 0   | 0  | 0   | 0  | 0   | 0  | 0   |
|         | Function    | Transmit data bit 8 | Handshake function control<br>0: CTS disable<br>1: CTS enable | Receive control<br>0: Reception disabled<br>1: Reception enabled | Wake-up function<br>0: Reception disabled<br>1: Reception enabled | Serial transfer mode<br>00: I/O interface mode<br>01: 7-bit length UART mode<br>10: 8-bit length UART mode<br>11: 9-bit length UART mode |     | Serial transfer clock (for UART)<br>00: TB4OUT<br>01: Baud rate generator<br>10: Internal clock f <sub>sys</sub><br>11: External clock (SCLK0 input) |     |

<TB8>: Writes the 9<sup>th</sup> bit of transmit data in the 9 bits UART mode.

<CTSE>: Controls handshake function.  
Setting "1" enables handshake function using  $\overline{\text{CTS}}$  pin.

<RXE>: Controls reception (**see note**).  
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.

## 10.4.5 Mode control register 1

|             | 7                           | 6   | 5     | 4   | 3  | 2     | 1     | 0          |
|-------------|-----------------------------|---|-------|---|--|-------|-------|------------|
| bit Symbol  | I2S0                        | FDPX1   | FDPX0 | TXE   | SINT2  | SINT1 | SINT0 | -          |
| 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<br>0: Stop<br>1: Start | Transfer mode setting<br>00: Transfer prohibited<br>01: Half duplex(RX)<br>10: Half duplex(TX)<br>11: Full duplex |       | Transmit control<br>0: Disabled<br>1: Enabled | Interval time of continuous transmission<br><b>(for I/O interface)</b><br>000: None 100: 8SCLK<br>001: 1SCLK 101: 16SCLK<br>010: 2SCLK 110: 32SCLK<br>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**).

<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)** Specify the mode first and then specify the <TXE> bit.

10.4.6 Mode control register 2

|             |   |   |   |   |  |                                       |  |        |
|-------------|---|---|---|---|--|---------------------------------------|--|--------|
|             | 7   | 6   | 5   | 4   | 3  | 2                                     | 1  | 0      |
| bit Symbol  | TBEMP   | RBFL  | TXRUN                                       | SBLN  | DRCHG  | WBUF                                  | SWRST1   | SWRST0 |
| Read/Write  | R   |   |   | R/W   |  |                                       |  |        |
| After reset | 1   | 0   | 0   | 0   | 0  | 0                                     | 0  | 0      |
| Function    | Transmit buffer empty flag<br>0: full<br>1: Empty | Receive Buffer full flag<br>0: Empty<br>1: full | In transmission flag<br>0: Stop<br>1: Start | STOP bit (for UART)<br>0: 1-bit<br>1: 2-bit | Setting transfer direction<br>0: LSB first<br>1: MSB first | W-buffer<br>0: Disabled<br>1: Enabled | SOFT RESET<br>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.

<RBFL>: 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 |

<SBLN>: 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 <SBLN> 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,RBFL, 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.

**10.4.7 Baud rate generator control register(SC0BRCR)  
Baud rate generator control register 2(SC0BRADD)**

|         |             |            |   |   |   |       |       |       |       |
|---------|-------------|------------|---|---|---|-------|-------|-------|-------|
|         |             | 7          | 6   | 5   | 4   | 3     | 2     | 1     | 0     |
|         | bit Symbol  | -          | BR0ADDE   | BR0CK1  | BR0CK0  | BR0S3 | BR0S2 | BR0S1 | BR0S0 |
|         | Read/Write  | R/W        |   |   |   |       |       |       |       |
| SC0BRCR | After reset | 0          | 0   | 0   | 0   | 0     | 0     | 0     | 0     |
|         | Function    | Write "0": | $N + (16 - K)/16$ divider function<br>0: disabled<br>1: enabled | Select input clock to the baud rate generator<br>00: $\phi T1$<br>01: $\phi T4$<br>10: $\phi T16$<br>11: $\phi T64$ | Division ratio "N"<br>0000: 16<br>0001: 1<br>0010: 2<br>:<br>1111: 15 |       |       |       |       |

|          |             |              |   |   |   |   |       |       |       |
|----------|-------------|--------------|---|---|---|---|-------|-------|-------|
|          |             | 7            | 6 | 5 | 4 | 3   | 2     | 1     | 0     |
|          | bit Symbol  |              |   |   |   | BR0K3   | BR0K2 | BR0K1 | BR0K0 |
|          | Read/Write  | R            |   |   |   | R/W   |       |       |       |
| SC0BRADD | After reset | 0            |   |   |   | 0   | 0     | 0     | 0     |
|          | Function    | "0" is read. |   |   |   | Specify K for the " $N + (16 - K)/16$ " division<br>0000: Prohibited<br>0001: K=1<br>0010: K=2<br>:<br>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 10-5 lists the settings of baud rate generator division ratio.

Table 10-5 Setting division ratio

|                |                               |                                    |
|----------------|-------------------------------|------------------------------------|
|                | BR0ADDE=0                     | BR0ADDE=1 (Note 1)<br>(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.

10.4.8 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<br>0: Maximum<br>1: Same as FILL level of RX FIFO | TX interrupt for TX FIFO<br>0: Disabled<br>1: Enabled | RX interrupt for RX FIFO<br>0: Disabled<br>1: Enabled | Automatic disable of RXE/TXE<br>0: None<br>1: Auto disable | FIFO enable<br>0: Disabled<br>1: Enabled |

- <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.



10.4.9 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<br>1: Clear<br>"0" is read. | Select interrupt generation condition<br>0: when the data reaches to the specified fill level.<br>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<br>00:4byte (2 Byte at full duplex)<br>01:1byte<br>10:2byte<br>11:3byte |      |

<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).

|    | 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.

10.4.10 TX FIFO configuration register

|             |  |  |              |   |   |   |  |      |
|-------------|--|--|--------------|---|---|---|--|------|
|             | 7  | 6  | 5            | 4 | 3 | 2 | 1  | 0    |
| bit Symbol  | TFCS                                       | TFIS   |              |   |   |   | TIL1   | TILO |
| Read/Write  | W  | R/W  | R            |   |   |   | R/W  |      |
| After reset | 0  | 0  | 0            |   |   |   | 0  | 0    |
| Function    | TX FIFO clear<br>1:Clear Always reads "0". | Select interrupt generation condition<br>0: when the data reaches to the specified fill level.<br>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.<br>00:Empty<br>01:1byte<br>10:2byte<br>11:3byte<br>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       |

- |                 |   |
|-----------------|---|
| <b>(Note 1)</b> | <b>TIL1 is ignored when FDPX1:0 = 11 (full duplex).</b>   |
| <b>(Note 2)</b> | <b>Please do clear transmission FIFO when FIFO is used after setting the forwarding mode of SIO, and permitting FIFO.</b>   |
| <b>(Note 3)</b> | <b>Please set SC0TFC again when shifting to standby mode (IDLE,STOP) with SC0EN&lt;0&gt;=0 (SIO operation prohibition * clock stop) or SC0MOD1&lt;7&gt;=0 (operation stop * clock stop of the standby mode inside).</b> |

## 10.4.11 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<br>Overflow<br><br>1:<br>Generated | "0" is read. |   |   |   | Status of RX FIFO fill level<br>000:Empty<br>001:1Byte<br>010:2Byte<br>011:3Byte<br>100:4Byte |       |       |

<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.

## 10.4.12 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<br>Under run<br>1:Generated<br>Cleared by<br>writing FIFO | "0" is read. |   |   |   | Status of TX FIFO fill level<br>000:Empty<br>001:1Byte<br>010:2Byte<br>011:3Byte<br>100:4Byte |       |       |

<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.

## 10.5 Operation in Each Mode

### 10.5.1 Mode 0 (I/O interface mode)

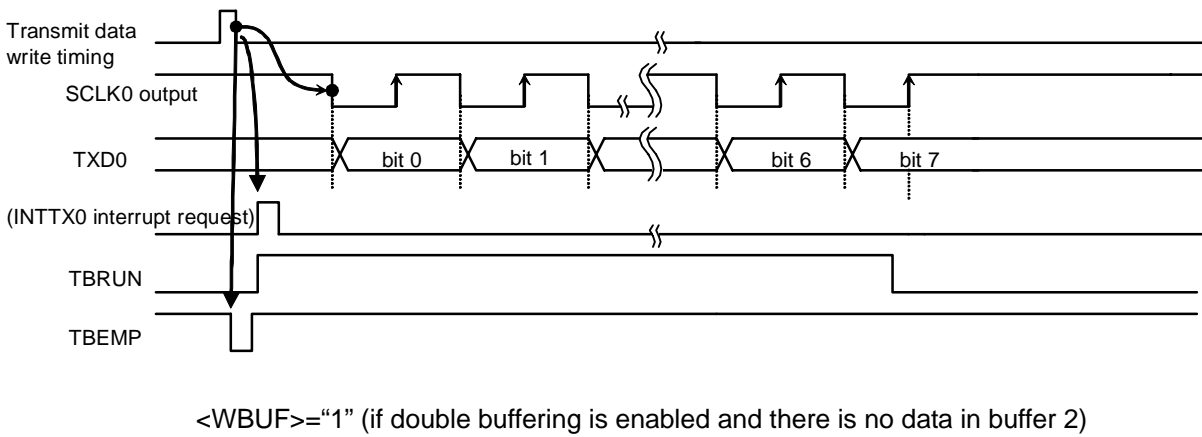
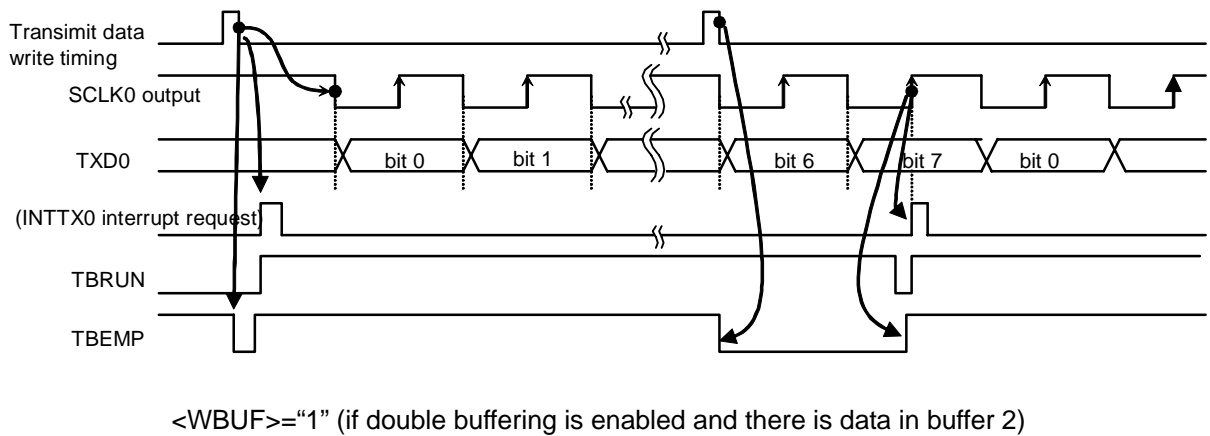
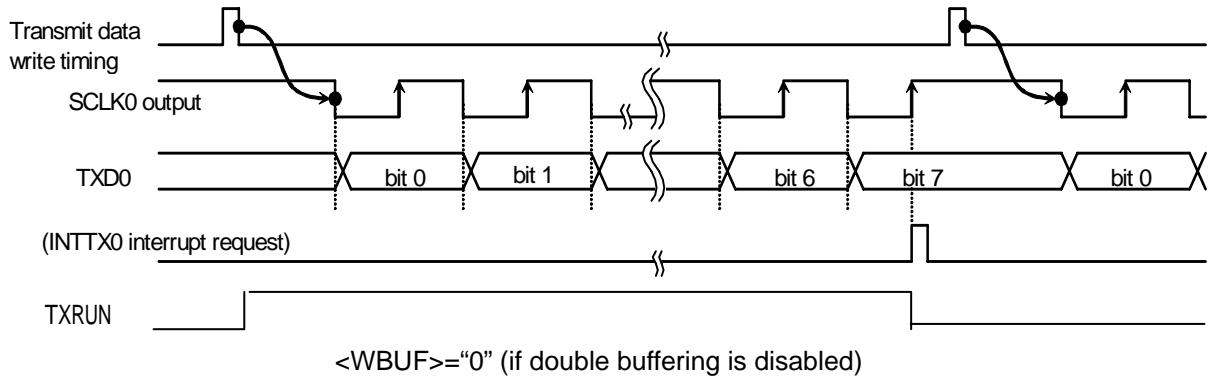
Mode 0 consists of two modes, the “SCLK output” mode to output synchronous clock and the “SCLK input” mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

① Transmitting data

SCLK output mode

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.

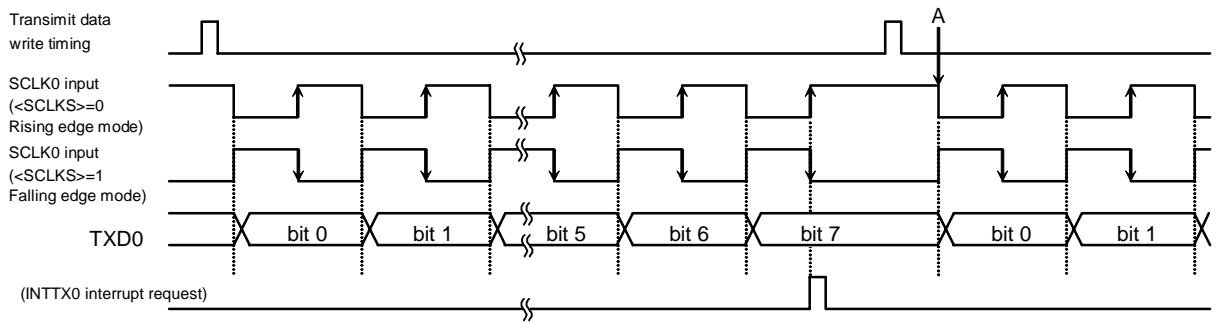


**Fig. 10-10 Transmit Operation in the I/O Interface Mode (SCLK0 Output Mode)**

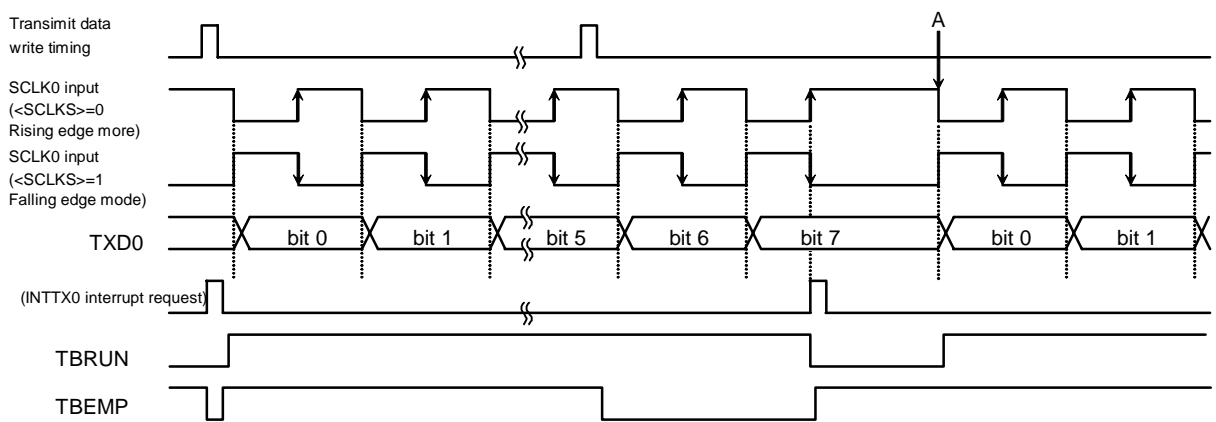
### SCLK input mode

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. 10-11.

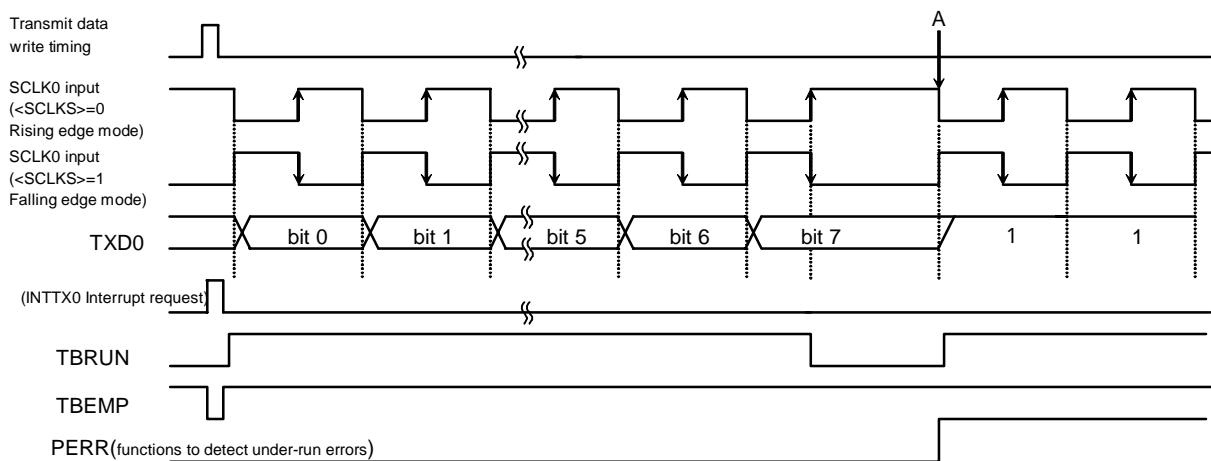
If SC0MOD2 <WBUF> is set to “1” and the transmit double buffers are enabled, data is moved from Transmit Buffer 2 to Transmit Buffer 1 when the CPU writes data to Transmit Buffer 2 before the SCLK0 becomes active or when data transmission from Transmit Buffer 1 (shift register) is completed. As data 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)

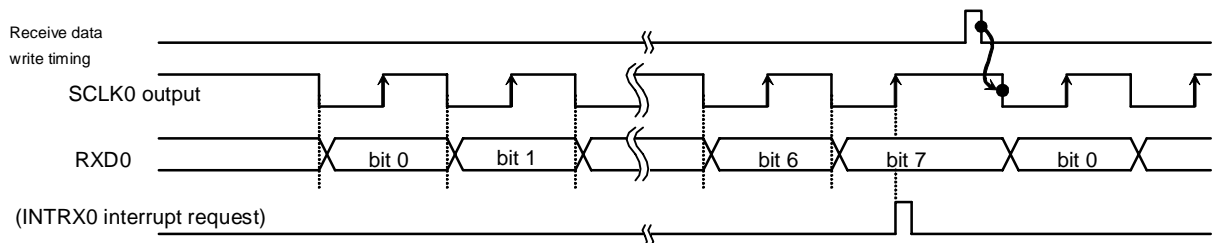
Fig. 10-11 Transmit Operation in the I/O Interface Mode (SCLK0 Input Mode)

② Receiving data  
SCLK output mode

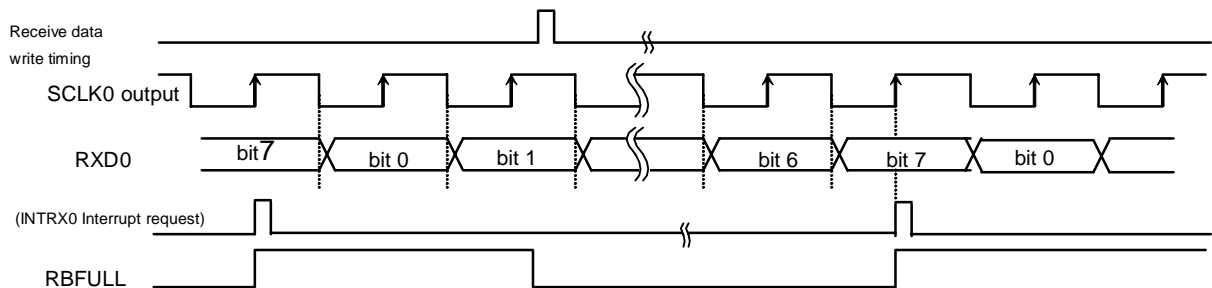
In the SCLK output mode, if SC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

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.

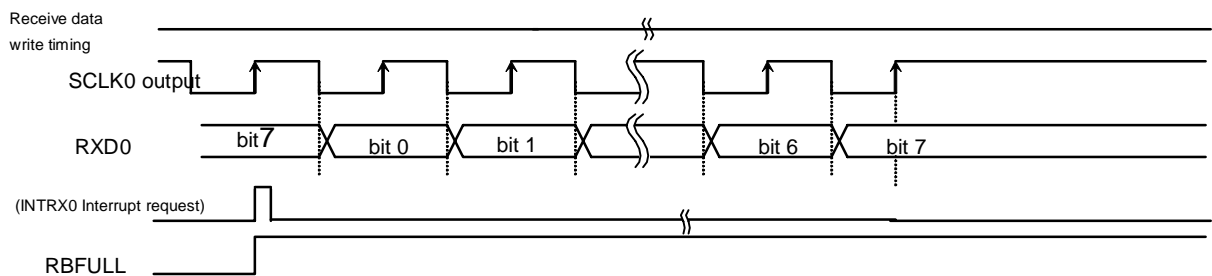
While data is in receive buffer 2, if CPU cannot read data from receive buffer 2 before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 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)

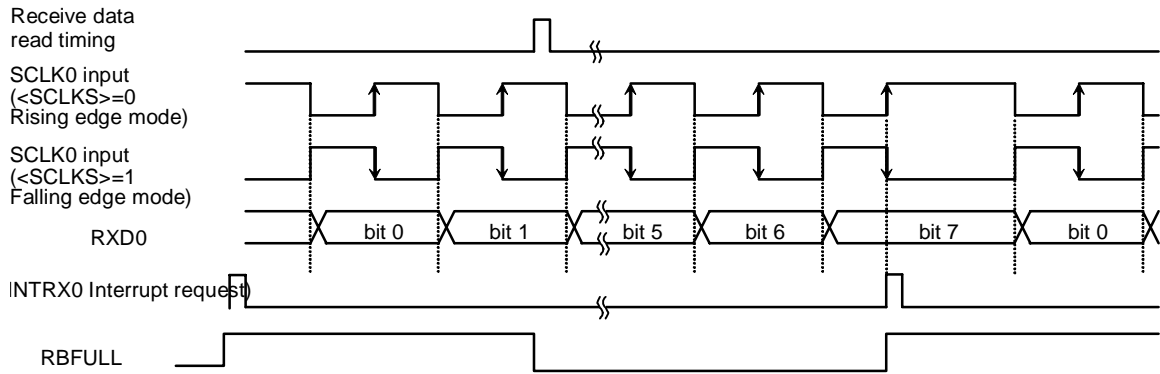
**Fig. 10-12 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)**



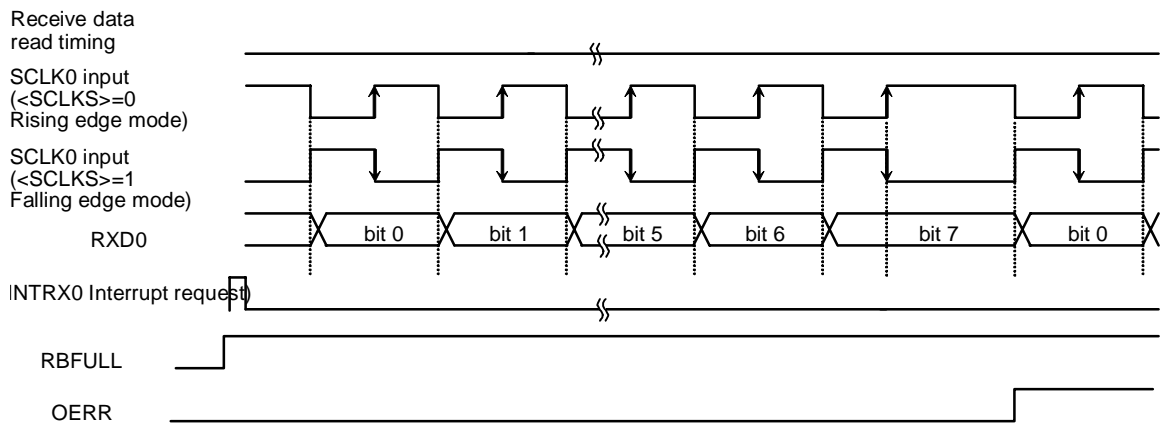
SCLK input mode

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. 10-13 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.

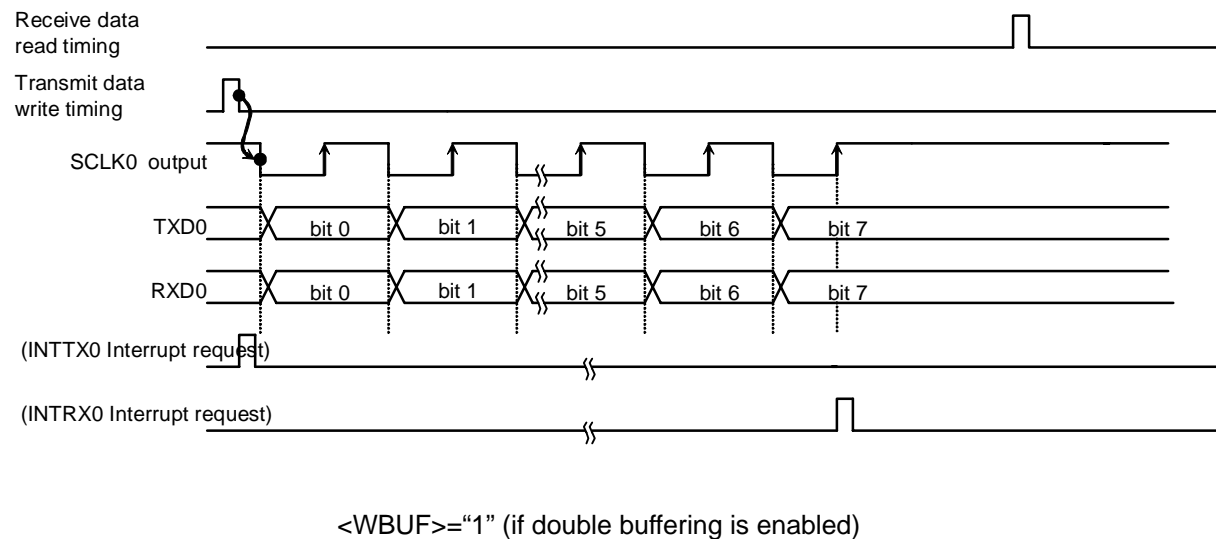
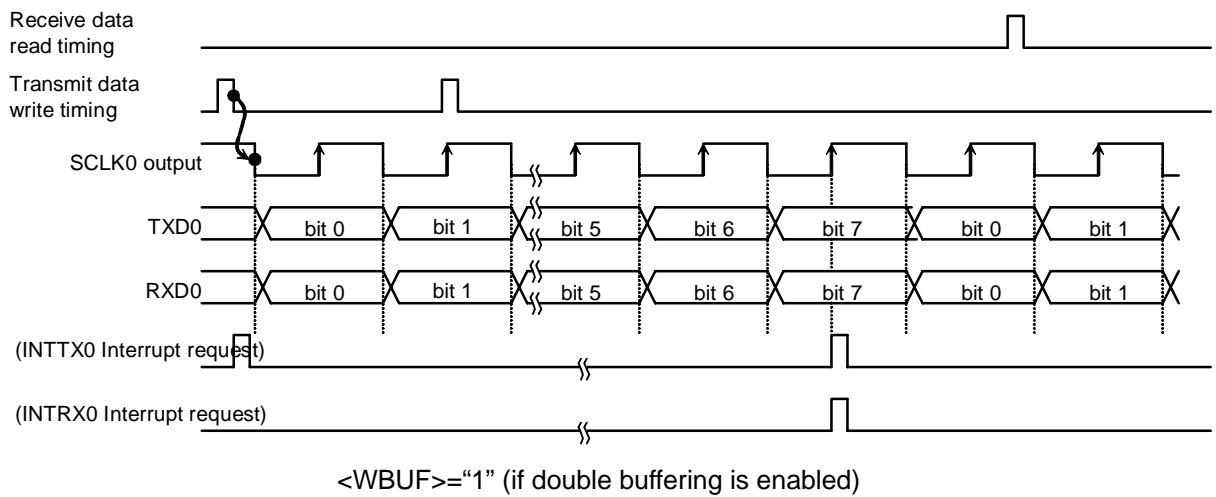
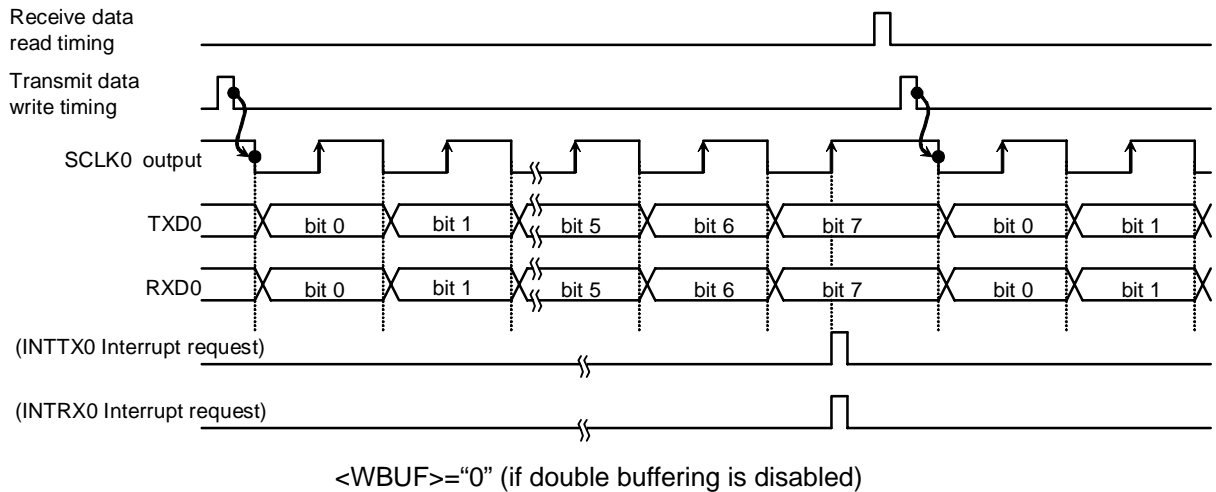
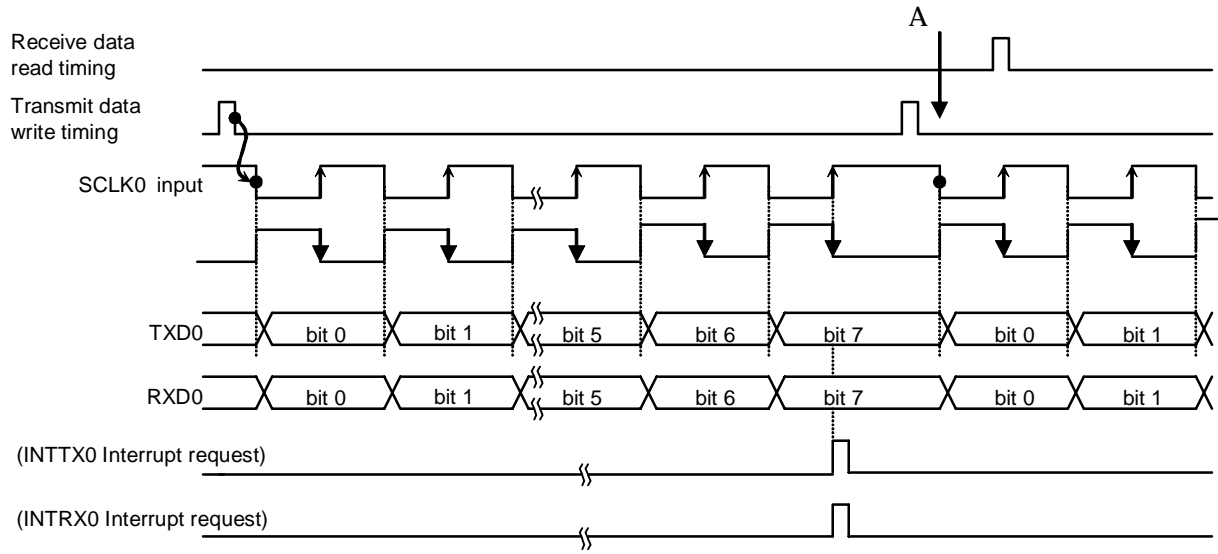


Fig. 10-14 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

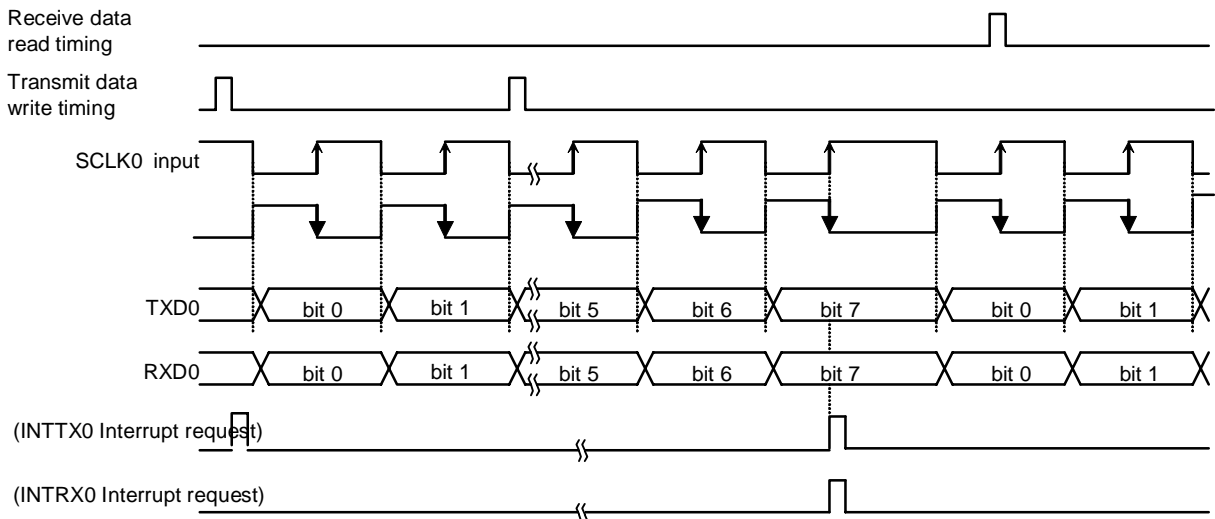
### SCLK input mode

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. 10-15). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

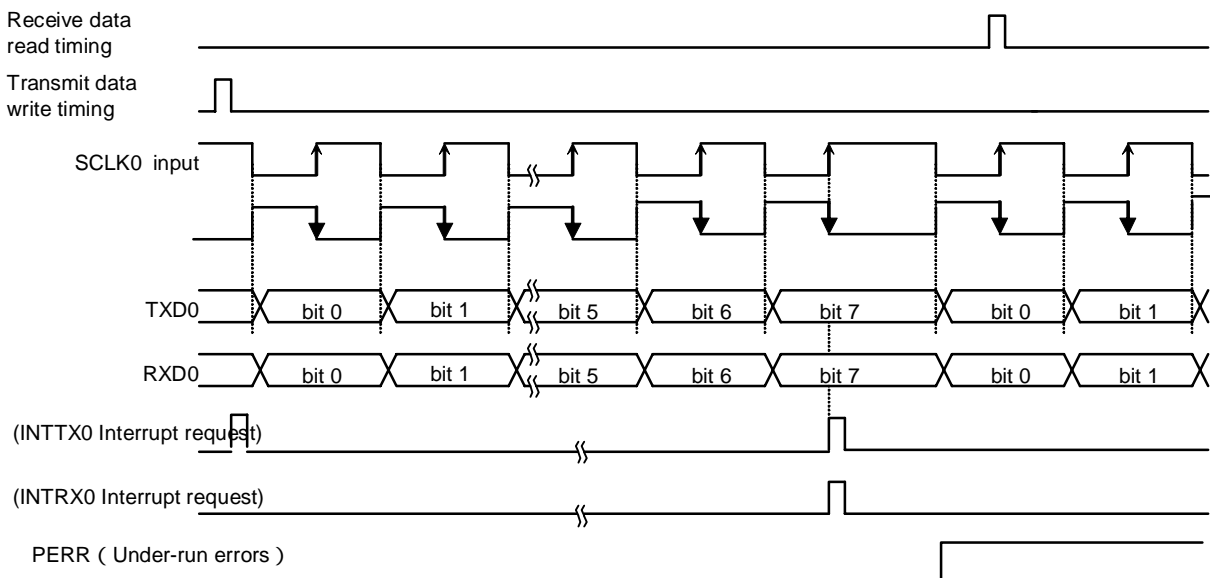
If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing Transmit Buffer 2 data is moved to Transmit Buffer 1 after completing data transmission from Transmit Buffer 1. At the same time, the 8 bits of data received is 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)

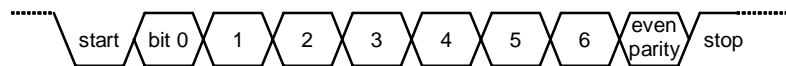
**Fig. 10-15 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)**

### 10.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SCOMOD <SM1, 0>) to "01".

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SCOCR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCOCR <EVEN> bit. The length of the stop bit can be specified using SCOMOD2<SBLLEN>.

The following table shows the control register settings for transmitting in the following data format.



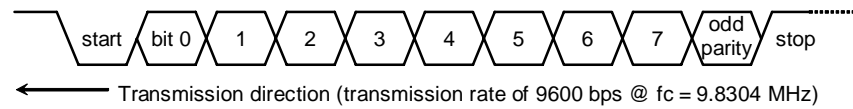
← Transmission direction (Transmission rate of 2400 bps @  $f_c = 9.8304$  MHz)

|                       |   |                       |   |  |
|-----------------------|---|-----------------------|---|--|
| * Clocking conditions | { | System clock          | : | high- speed ( $f_c$ )  |
|                       |   | High-speed clock gear | : | x1 ( $f_c$ )   |
|                       |   | Prescaler clock       | : | $f_{\text{periph}}/2$ ( $f_{\text{periph}} = f_{\text{sys}}$ ) |

### 10.5.3 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:



\* Clocking conditions

|                       |   |  |
|-----------------------|---|--|
| System clock          | : | High-speed (fc)  |
| High-speed clock gear | : | x1 (fc)  |
| Prescaler clock       | : | f <sub>periph</sub> /4 (f <sub>periph</sub> = f <sub>sys</sub> ) |

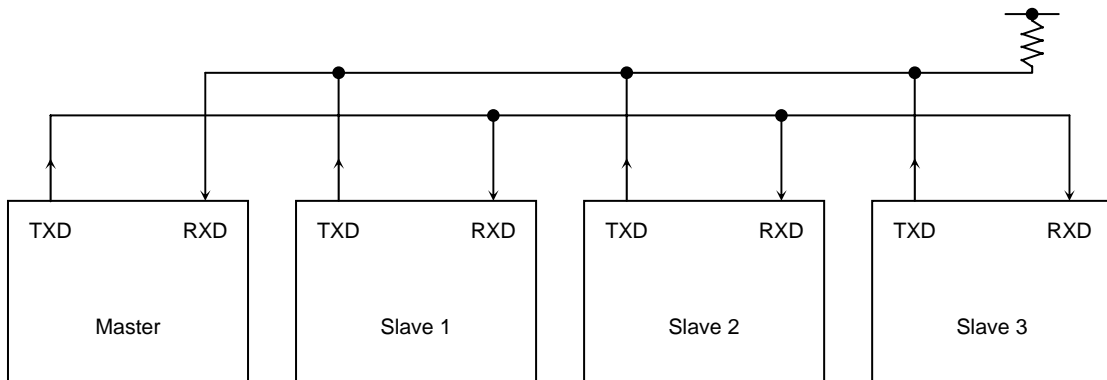
**10.5.4 Mode 3 (9-bit UART)**

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>.

Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SC0MOD0 <WU> to “1.” In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to “1”.



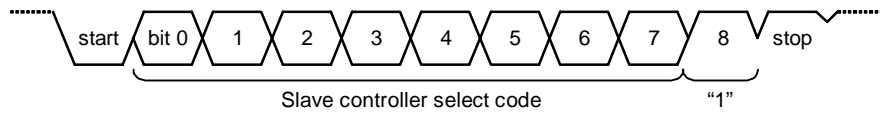
**(Note)The TXD pin of the slave controller must be set to the open drain output mode using the ODE register.**

**Fig. 10-16 Serial Links to Use Wake-up Function**

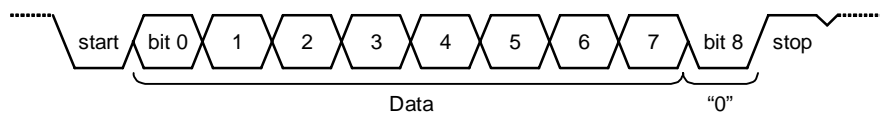


Protocol

- 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".

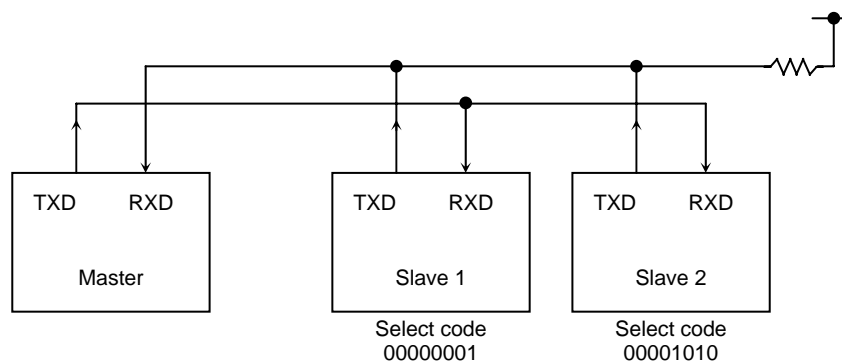


- ④ Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0".
- ⑤ The master controller transmits data to the designated slave controller (the controller of which SC0MOD <WU> bit is cleared to "0"). In this, the 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.



## 11 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.

### 11.1 Configuration

Fig. 11-1 shows the block diagram of the watchdog timer

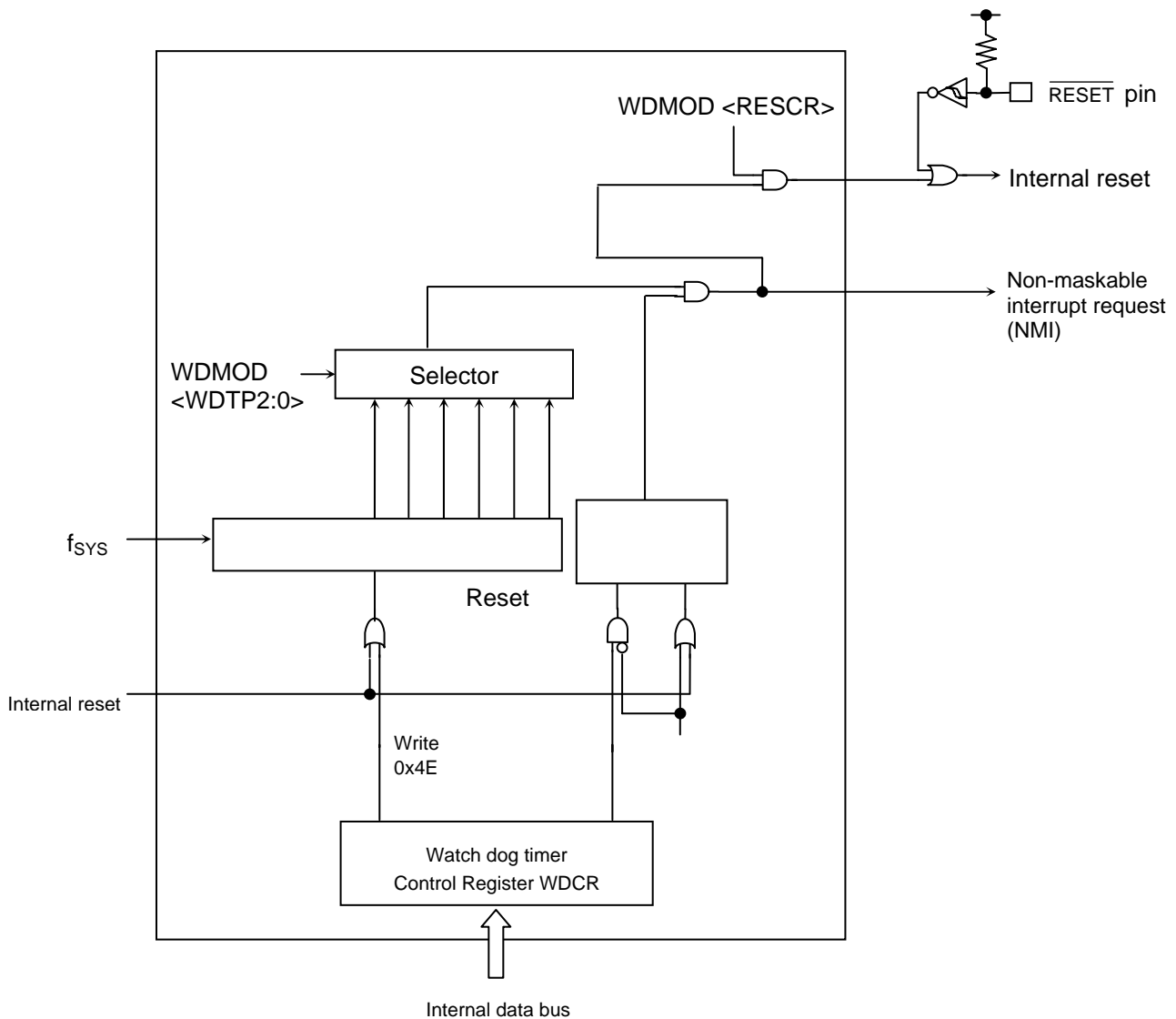


Fig. 11-1 Block Diagram of the Watchdog Timer

## 11.2 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. 11-2.

### 11.2.1 Interrupt Mode (WDMOD<RESCR>=0)

When an overflow occurs, the watchdog timer generates NMI to the CPU.

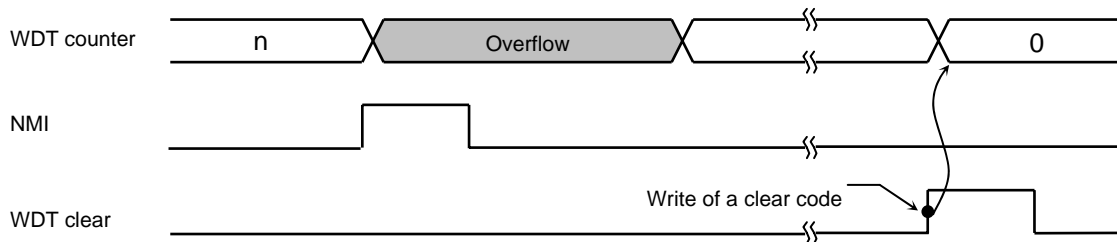


Fig. 11-2 Interrupt Mode

### 11.2.2 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. 11-3. 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  $f_{SYS}$ .

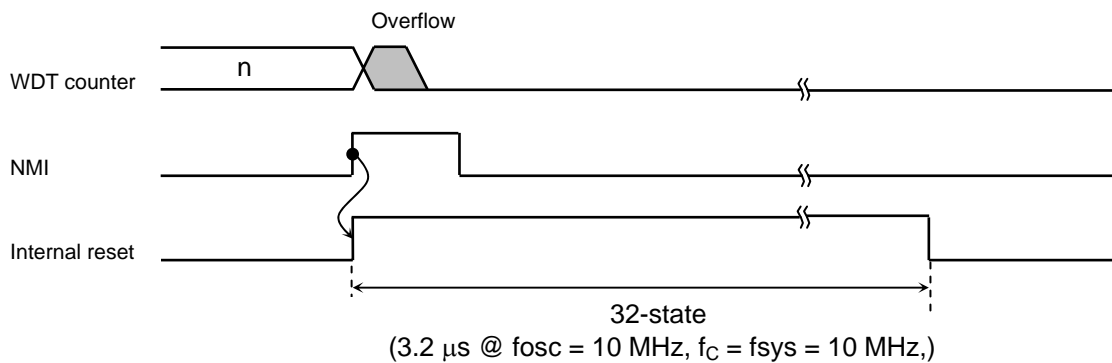


Fig. 11-3 Reset Mode

## 11.3 Control Registers

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

### 11.3.1 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. 11-4 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.

### 11.3.2 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

WDMOD  
(0x4004\_0000)

|             |  |  |       |       |              |                             |   |            |
|-------------|--|--|-------|-------|--------------|-----------------------------|---|------------|
|             | 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<br>0: Disable<br>1: Enable | Selects WDT detection time<br>000: $2^{15}/f_{SYS}$<br>001: $2^{17}/f_{SYS}$<br>010: $2^{19}/f_{SYS}$<br>011: $2^{21}/f_{SYS}$<br>100: $2^{23}/f_{SYS}$<br>101: $2^{25}/f_{SYS}$<br>110: Setting prohibited<br>111: Setting prohibited |       |       | "0" is read. | IDLE<br>0: Stop<br>1: Start | WDT Out control<br>0: Generates NMI<br>1: Generates reset | Write "0." |

Watchdog timer out control

|   |                 |
|---|-----------------|
| 0 | Generates NMI   |
| 1 | Generates reset |

Detection time of watchdog timer @  $f_c = 80 \text{ MHz}$

| SYSCR1<br>clock gear<br>value<br><GEAR2:0> | Detection time of watchdog timer |          |           |           |           |           |
|--|----------------------------------|----------|-----------|-----------|-----------|-----------|
|  | WDMOD<WDTP2:0>                   |          |           |           |           |           |
|  | 000                              | 001      | 010       | 011       | 100       | 101       |
| 000 ( $f_c$ )                              | 0.41 ms                          | 1.64 ms  | 6.55 ms   | 26.21 ms  | 104.21 ms | 419.43 ms |
| 100 ( $f_c/2$ )                            | 0.82 ms                          | 3.28 ms  | 13.11 ms  | 52.43 ms  | 209.72 ms | 838.86 ms |
| 101 ( $f_c/4$ )                            | 1.64 ms                          | 6.55 ms  | 26.21 ms  | 104.86 ms | 419.43 ms | 1.68 s    |
| 110 ( $f_c/8$ )                            | 3.28 ms                          | 13.11 ms | 52.43 ms  | 209.72 ms | 838.86 ms | 3.36 s    |
| 111 ( $f_c/16$ )                           | 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

WDCR  
(0x4004\_0004)

|             |  |   |   |   |   |   |   |   |
|-------------|--|---|---|---|---|---|---|---|
|             | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| bit Symbol  | -  |   |   |   |   |   |   |   |
| Read/Write  | W  |   |   |   |   |   |   |   |
| After reset | -  |   |   |   |   |   |   |   |
| Function    | 0xB1 : WDT disable code<br>0x4E : WDT clear code |   |   |   |   |   |   |   |

Disable & clear of WDT

|        |                  |
|--------|------------------|
| 0xB1   | WDT disable code |
| 0x4E   | WDT clear code   |
| Others | -                |

Fig. 11-4 Watchdog Timer Registers

## 11.4 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.**

## 12. Oscillation Frequency Detector (OFD)

### 12.1 Configuration

The oscillation frequency detector generates a reset for I/O if the oscillation of high frequency for CPU clock OFDMNPLL exceeds the detection frequency range.

The oscillation frequency detection is controlled by OFDCR1, OFDCR2 registers and the detection frequency range is specified by OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON which are the detection frequency setting registers. The lower detection frequency is specified by OFDMNPLLOFF/OFDMNPLLON registers and the higher detection frequency is specified by OFDMXPLLOFF/OFDMXPLLON registers. An initial value of detection frequency is shown in Figure 12-1.

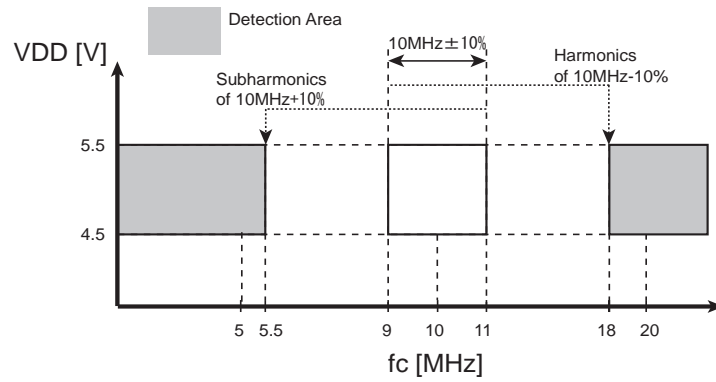
When the oscillation frequency detection is enabled, writing to OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers is disabled. Therefore, the setting the detection frequency to these registers should be done when the oscillation frequency detection is disabled. And writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers is controlled by OFDCR1 register. To write OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers, the write enable code "0xF9" should be set to OFDCR1 beforehand. To enable the oscillation frequency detector, set "0xE4" to OFDCR2 after setting "0xF9" to OFDCR1. Since the oscillation frequency detection is disabled after an external reset input, write "0xF9" to OFDCR1 and write "0xE4" to OFDCR2 register to enable its function.

When the TMPM370 detects the out of frequency by lower and higher detection frequency setting registers, all I/Os become high impedance by reset. In case of PLLOFF, OFDMNPLLOFF and OFDMXPLLOFF registers are valid for detection and the setting value of OFDMNPLLON/OFDMXPLLON registers are ignored. In case of PLLON, OFDMNPLLON and OFDMXPLLON registers are valid for detection and the setting value of OFDMNPLLOFF/OFDMXPLLOFF registers are ignored. By the oscillation frequency detection reset, all I/Os except power supply pins, RESET, X1 and X2 become high impedance. If oscillation frequency detection reset is generated by detecting the stopping of high frequency, the internal circuitries 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.

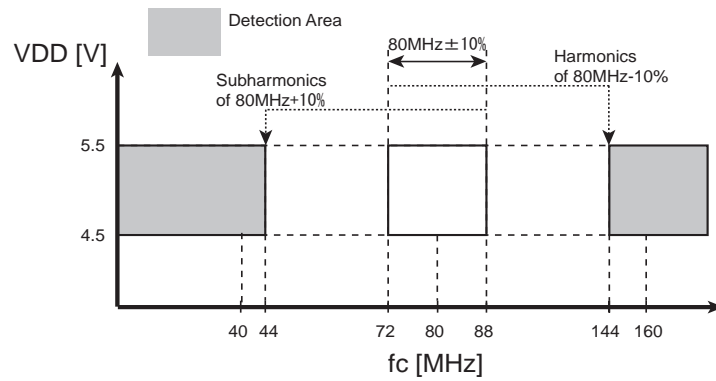
Since all registers for oscillation frequency detector (OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON) are not initialized by the reset generated from oscillation frequency detector, the detection of oscillation keeps its function during the reset period of oscillation frequency detection. 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: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.



(a) In case of PLL OFF



(b) In case of PLL ON

Figure 12-1 Detection frequency range (Initial value)

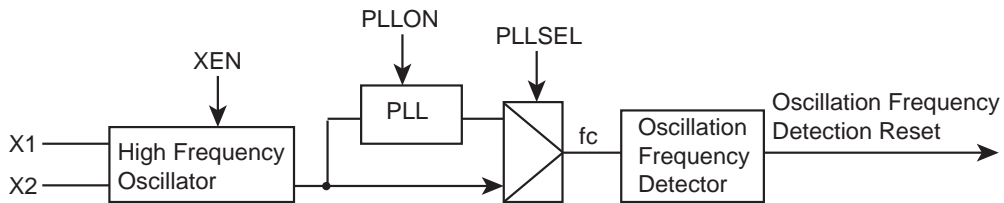


Figure 12-2 Oscillation Frequency Detector



## 12.2 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/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON is controlled by oscillation frequency control register 1 (OFDCR1).

### Oscillation frequency detection control register 1

|                         |   |         |         |         |         |         |         |         |   |
|-------------------------|---|---------|---------|---------|---------|---------|---------|---------|---|
| OFDCR1<br>(0x4004_0800) | 31-8  |         |         |         |         |         |         |         |   |
|                         | Bit Symbol  | -       |         |         |         |         |         |         |   |
|                         | 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)<br>0xF9: Enabling of writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON (Write enable code)<br>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 not initialized by an internal factor reset including oscillation frequency detection reset. To initialize this registers, set the RESET pin (external factor reset) to the low level.

### Oscillation frequency detection control register 2

|                         |  |         |         |         |         |         |         |         |   |
|-------------------------|--|---------|---------|---------|---------|---------|---------|---------|---|
| OFDCR2<br>(0x4004_0804) | 31-8   |         |         |         |         |         |         |         |   |
|                         | Bit Symbol   | -       |         |         |         |         |         |         |   |
|                         | 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<br>0xE4: Enabling of oscillation frequency detection<br>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 not initialized by an internal factor reset including oscillation frequency detection reset. To initialize this registers, set the RESET pin (external factor reset) to the low level.

Lower detection frequency setting register (In case of PLL OFF)

|                              |             |             |     |     |     |     |     |             |     |
|------------------------------|-------------|-------------|-----|-----|-----|-----|-----|-------------|-----|
|                              | 31-9        |             |     |     |     |     | 8   |             |     |
| OFDMNPLLOFF<br>(0x4004_0808) | Bit Symbol  | -           |     |     |     |     |     | OFDMNPLLOFF |     |
|                              | 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   | 1   | 1           | 1   |

Lower detection frequency setting register (In case of PLL ON)

|                             |             |            |     |     |     |     |     |            |     |
|-----------------------------|-------------|------------|-----|-----|-----|-----|-----|------------|-----|
|                             | 31-9        |            |     |     |     |     | 8   |            |     |
| OFDMNPLLON<br>(0x4004_080C) | Bit Symbol  | -          |     |     |     |     |     | OFDMNPLLON |     |
|                             | 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   | 1   | 1   | 0   | 0   | 0          | 1   |

Higher detection frequency setting register (In case of PLL OFF)

|                              |             |             |     |     |     |     |     |             |     |
|------------------------------|-------------|-------------|-----|-----|-----|-----|-----|-------------|-----|
|                              | 31-9        |             |     |     |     |     | 8   |             |     |
| OFDMXPLLOFF<br>(0x4004_0810) | Bit Symbol  | -           |     |     |     |     |     | OFDMXPLLOFF |     |
|                              | 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   | 1   | 0   | 1   | 1           | 1   |

Higher detection frequency setting register (In case of PLL ON)

|                             |             |            |     |     |     |     |     |            |     |
|-----------------------------|-------------|------------|-----|-----|-----|-----|-----|------------|-----|
|                             | 31-9        |            |     |     |     |     | 8   |            |     |
| OFDMXPLLON<br>(0x4004_0814) | Bit Symbol  | -          |     |     |     |     |     | OFDMXPLLON |     |
|                             | Read/Write  | R          |     |     |     |     |     | R/W        |     |
|                             | After reset | 0          |     |     |     |     |     | 0          |     |
|                             |             | 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 | 1          | 1   | 0   | 0   | 0   | 0   | 0          | 0   |

Note 1: 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 OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON can not complete a write operation.

Note 2: 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 3: 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 " 12.3.2 Setting the Lower and Higher Frequency for Detection ".

- Note 4: 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 " 12.3.2 Setting the Lower and Higher Frequency for Detection ".
- Note 5: OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON are not initialized by an internal factor reset including oscillation frequency detection reset. To initialize these registers, set the RESET pin (external factor reset) to the low level.
- Note 6: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

## 12.3 Function

### 12.3.1 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.

Setting "0xF9" to OFDCR1 enables writing to OFDCR2 and setting "0x06" to OFDCR1 disables writing to OFDCR2. Reading from OFDCR2 is always enabled without setting of OFDCR1. OFDCR1 is initialized to "0x06" by external reset and OFDCR2 is initialized to "0x00" by external reset. However, OFDCR1 and OFDCR2 are not initialized by internal reset which are SYSRESETREQ reset, watchdog timer reset and oscillation frequency detection reset.

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 12-1 shows the availability of oscillation frequency detector.

Table 12-1 Availability of oscillation frequency detector

| Operating Mode                                 | Oscillation Frequency Detection (OFDCR2=0xE4)              | All I/Os condition after Oscillation Frequency Detection RESET (Except power supply, 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   |
| RESET by internal reset (Note 1)               | Available  | High impedance   |
| RESET by external reset                        | Disable  | -  |

Note 1: Internal reset; Watchdog timer reset, SYSRESETREQ reset

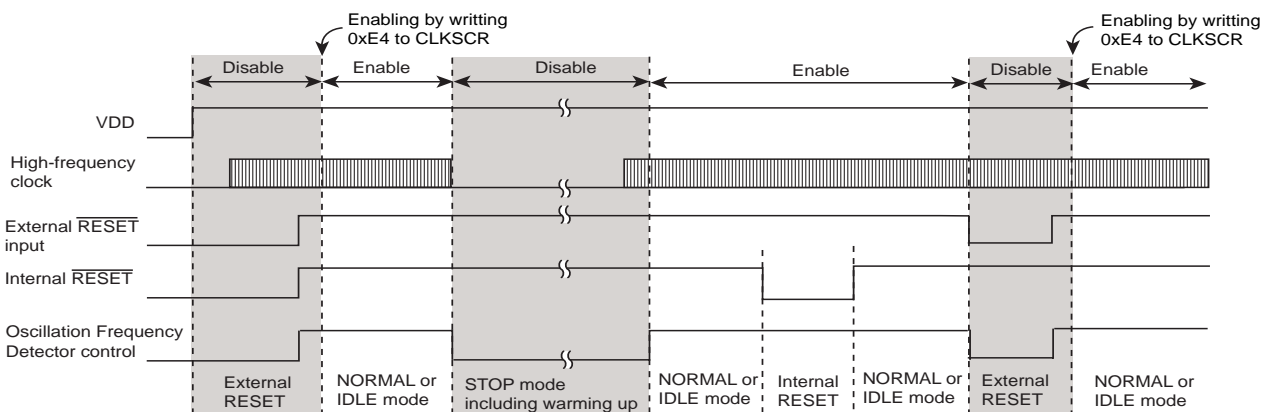


Figure 12-3 Availability of Oscillation Frequency Detection

### 12.3.2 Setting the Lower and Higher Frequency for Detection

The detection frequency is specified by OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON registers. The relation between the setting value to these registers and the detection frequency is shown in Table 12-2.

Table 12-2 High frequency and Setting value for detection frequency

| High frequency     | Range | Detection frequency range [MHz] | Non detection frequency range [MHz] | Setting value Decimal (Hex) |
|--------------------|-------|---------------------------------|-------------------------------------|-----------------------------|
| 8MHz<br>(PLL OFF)  | Lower | $\leq 4.12$                     | $\geq 7.76$                         | OFDMNPLLOFF 24 (0x18)       |
|                    | Upper | $\geq 15.52$                    | $\leq 8.24$                         | OFDMXPLLOFF 47 (0x2F)       |
| 10MHz<br>(PLL OFF) | Lower | $\leq 5.15$                     | $\geq 9.7$                          | OFDMNPLLOFF 30 (0x1E)       |
|                    | Upper | $\geq 19.4$                     | $\leq 10.3$                         | OFDMXPLLOFF 59 (0x3B)       |
| 64MHz<br>(PLL ON)  | Lower | $\leq 32.96$                    | $\geq 62.08$                        | OFDMNPLLON 190 (0xBE)       |
|                    | Upper | $\geq 124.16$                   | $\leq 65.92$                        | OFDMXPLLON 379 (0x17B)      |
| 80MHz<br>(PLL ON)  | Lower | $\leq 41.2$                     | $\geq 77.6$                         | OFDMNPLLON 238 (0xEE)       |
|                    | Upper | $\geq 155.2$                    | $\leq 82.4$                         | OFDMXPLLON 473 (0x1D9)      |

### 12.3.3 Oscillation Frequency Detection Reset

If the TMPM370 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

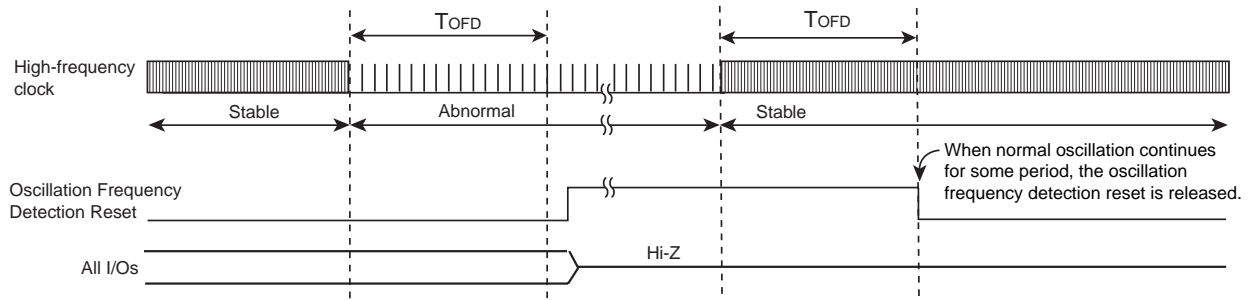
When an abnormal (lower or higher) frequency oscillation continues for some period ( $T_{OFD}$ ), the oscillation frequency detection reset is generated. By oscillation frequency detection reset initializes all I/Os except power supply pins, RESET, X1 and X2 become high impedance.

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, 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

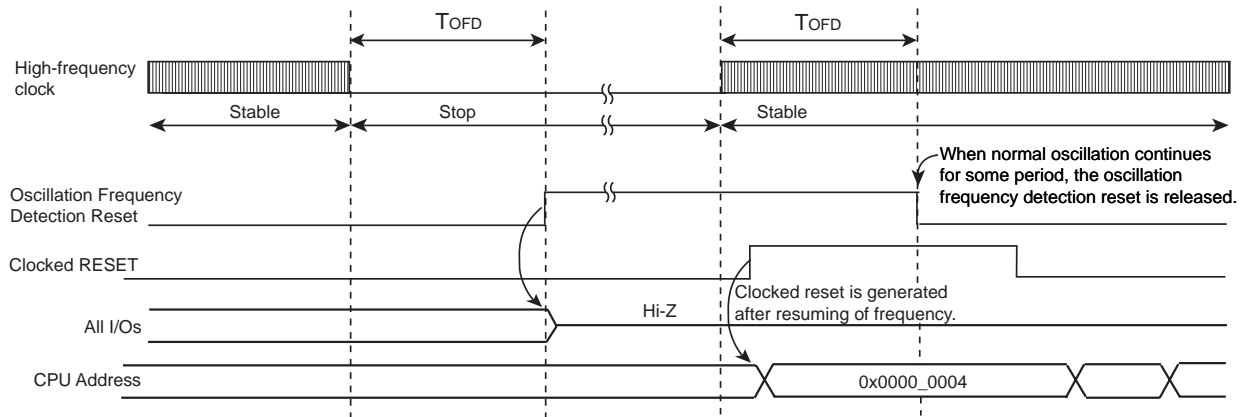


Figure 12-4 Oscillation Frequency Detection Reset Timing



## 13. 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.

### 13.1 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.

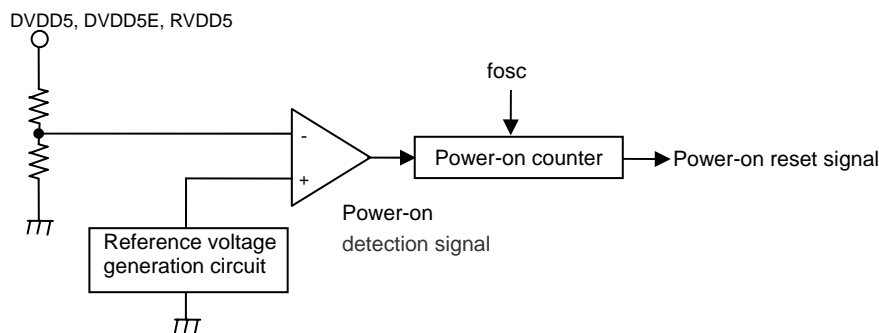


Figure 13-1 Power-on Reset Circuit

### 13.2 Function

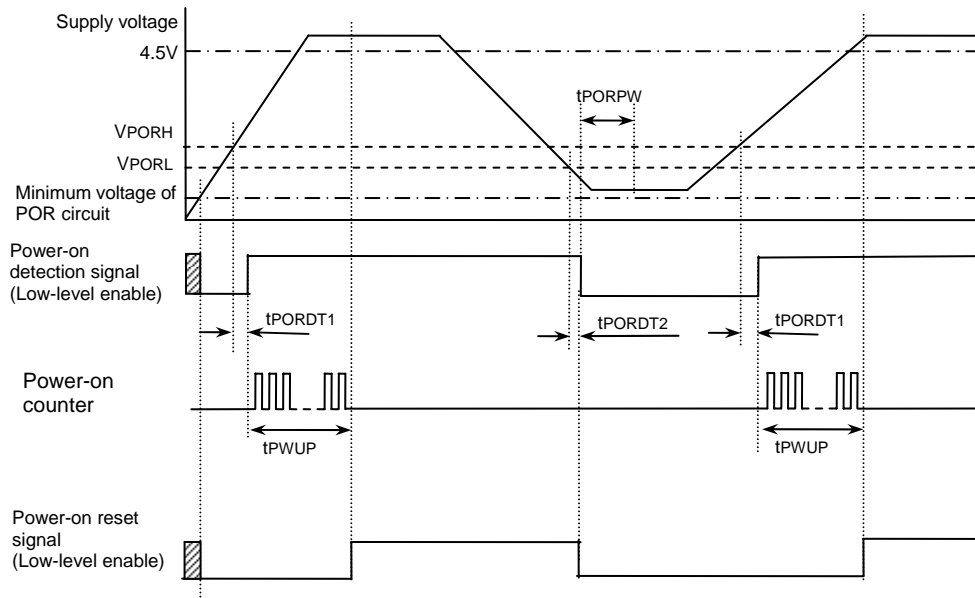
When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated. If the power supply voltage exceeds the releasing voltage of the power-on reset circuit, power-on counter is activated and  $2^{15}/f_{osc}$  (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 circuit 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 TMPM370 cannot operate properly.

(Note) When the supply voltage rises, until the supply voltage (at DVDD5, DVDD5E and RVDD5 pins) reach the recommended operating voltage range (4.5V through 5.5V) and 200 $\mu$ s passes by, the following condition should be satisfied; Port L (PL0 and PL1) is opened or the input voltage is within 0.5 volts.



- 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 13-2 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)              |     | $2^{15}/f_{osc}$ |     | 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”.



## 14. 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 (DVDD5, DVDD5E, RVDD5). Refer to the electrical characteristics and take them into consideration when designing equipment.

### 14.1 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 (DVDD5, DVDD5E and 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 (DVDD5, DVDD5E and RVDD5) becomes lower than the detection voltage (VDLVL), a voltage detection reset signal is generated.

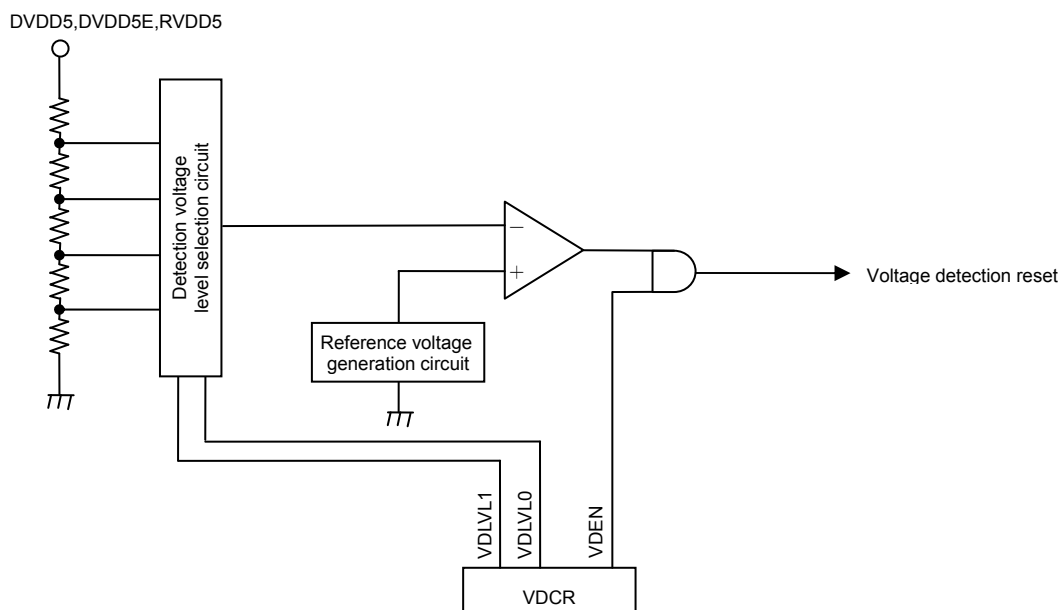


Figure 14-1 Voltage Detection Circuit

## 14.2 Control

The voltage detection circuit is controlled by voltage detection control registers.

Voltage detection control register

|                       |             |   |   |   |   |   |        |        |      |
|-----------------------|-------------|---|---|---|---|---|--------|--------|------|
| VDCR<br>(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<br>01 : $4.1 \pm 0.2$ V<br>10 : $4.4 \pm 0.2$ V<br>11 : $4.6 \pm 0.2$ V             |
| VDEN       | Enables/disables the operation of voltage detection | 0 : Disables the operation of voltage detection<br>1 : Enables the operation of voltage detection |

Note 1: VDCR is initialized by a power-on reset or an external reset input.

## 14.3 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 (DVDD5, DVDD5E and RVDD5) becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection reset signal is generated.

### 14.3.1 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 (DVDD5, DVDD5E and RVDD5) is lower than the detection voltage (VDLVL), setting VDCR<VDEN> to "1" generates reset signal at the time.

### 14.3.2 Selecting the detection voltage level

Select a detection voltage at  $VDCR<VDLVL[1:0]>$ .

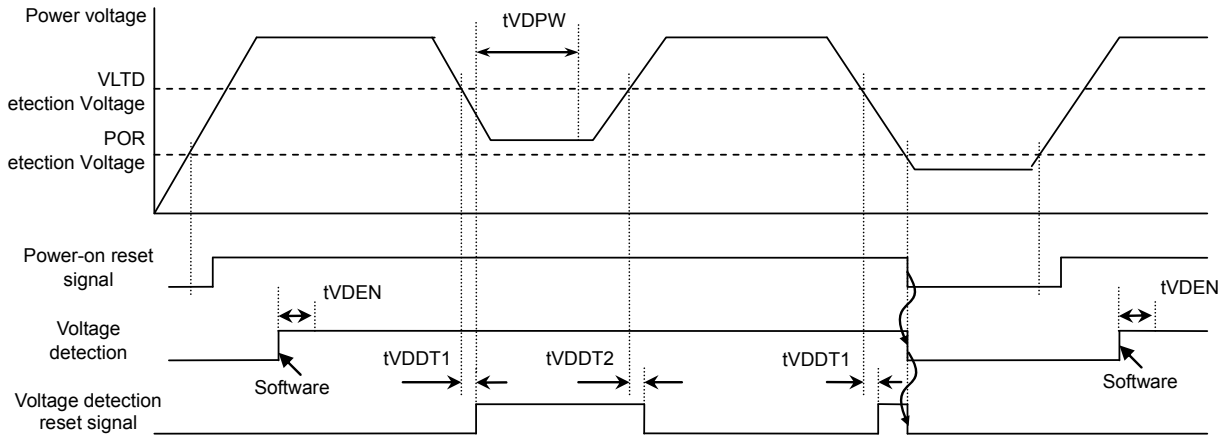


Figure 14-2 Voltage Detection Timing

| Symbol      | Parameter                                   | Min | Typ. | Max | Unit    |
|-------------|---|-----|------|-----|---------|
| $t_{VDEN}$  | Setup time after enabling voltage detection |     | 40   |     | $\mu s$ |
| $t_{VDDT1}$ | Voltage detection response time             |     | 40   |     | $\mu s$ |
| $t_{VDDT2}$ | Voltage detection releasing time            |     | 40   |     | $\mu s$ |
| $t_{VDPW}$  | Voltage detection minimum pulse width       | 45  |      |     | $\mu s$ |

## 15. Vector Engine (VE)

### 15.1 Overview

#### 15.1.1 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).

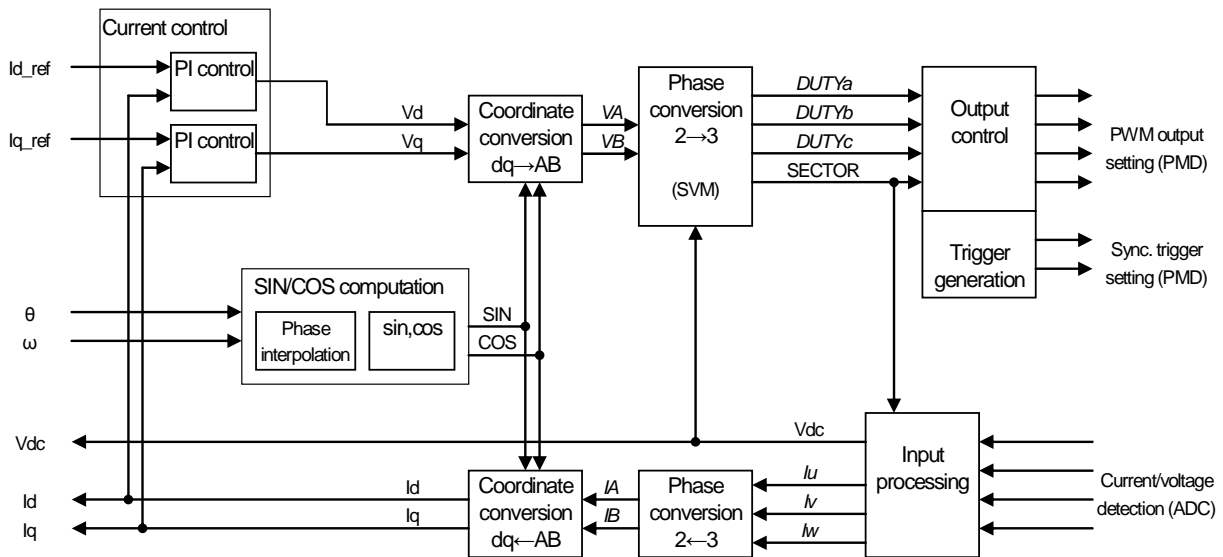


Figure 15-1 Block Diagram of Vector Control

### 15.1.2 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 (MODESEL).  
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.

## 15.2 Configuration

Figure 15-2 shows the configuration of the Vector Engine.

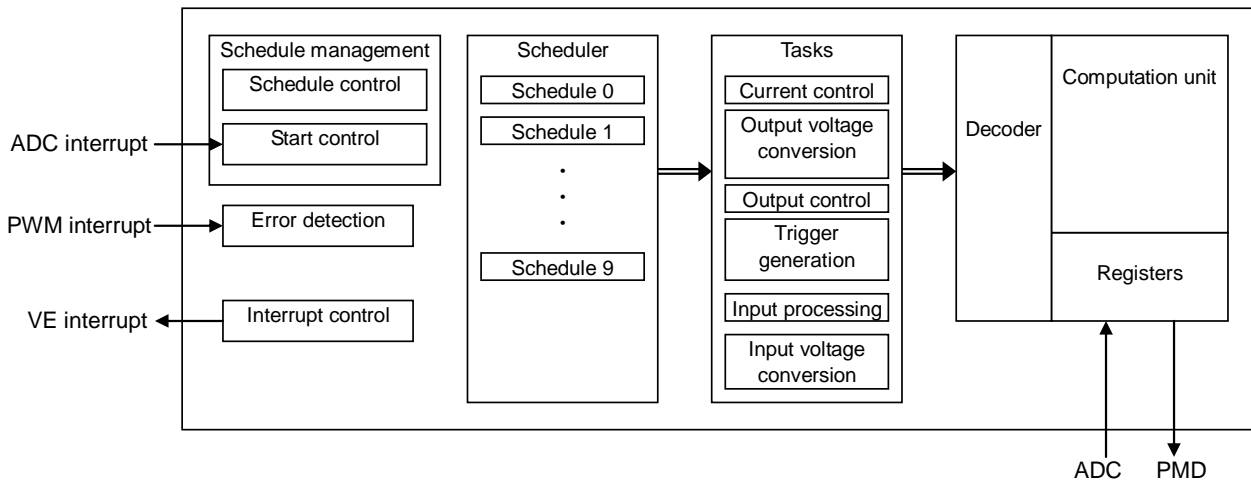


Figure 15-2 Configuration of the Vector Engine

### 15.2.1 Interaction among Vector Engine, Motor Control Circuit and A/D Converter

The Vector Engine can control two motors by interacting with two channels of motor control circuit (PMD) and two units of AD converter (ADC). Channel 0 of the Vector Engine controls PMD channel 0, and channel 1 of the Vector Engine controls PMD channel 1.

As shown in Figure 15-3, the Vector Engine allows direct interaction with the PMD and ADC.

When the PMD0MODESEL register is set to the VE mode, the PMD channel 0 registers PMD0CMPU, PMD0CMPV, PMD0CMPW, PMD0MDOUT, PMD0TRGCMP0, PMD0TRGCMP1 and PMD0TRGSEL are switched to the Vector Engine registers VECMPU0, VECMPV0, VECMPW0, VEOUTCR0, VETRGCMP00, VETRGCMP10 and VETRGSEL0 respectively. Likewise, the PMD channel 1 registers are switched to Vector Engine registers VECMPU1, VECMPV1, VECMPW1, VEOUTCR1, VETRGCMP01, VETRGCMP11 and VETRGSEL1. 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 A registers ADAREG0, ADAREG1, ADAREG2, ADAREG3 and UVWISx0, UVWISx1, UVWISx2, UVWISx3 which are the phase information specified by ADAPSETx are read into the Vector Engine as the Vector Engine registers VEADREG0A, VEADREG1A, VEADREG2A, VEADREG3A, VEPHNUM0A, VEPHNUM1A, VEPHNUM2A and VEPHNUM3A respectively. (These registers cannot be accessed from the CPU.) Likewise, the ADC unit B registers are read into the Vector Engine as the Vector Engine registers VEADREG0B, VEADREG1B, VEADREG2B, VEADREG3B, VEPHNUM0B, VEPHNUM1B, VEPHNUM2B and VEPHNUM3B. (These registers cannot be accessed from the CPU.) These ADC registers can be written and read from the ADC.

(x = 0 to 5)

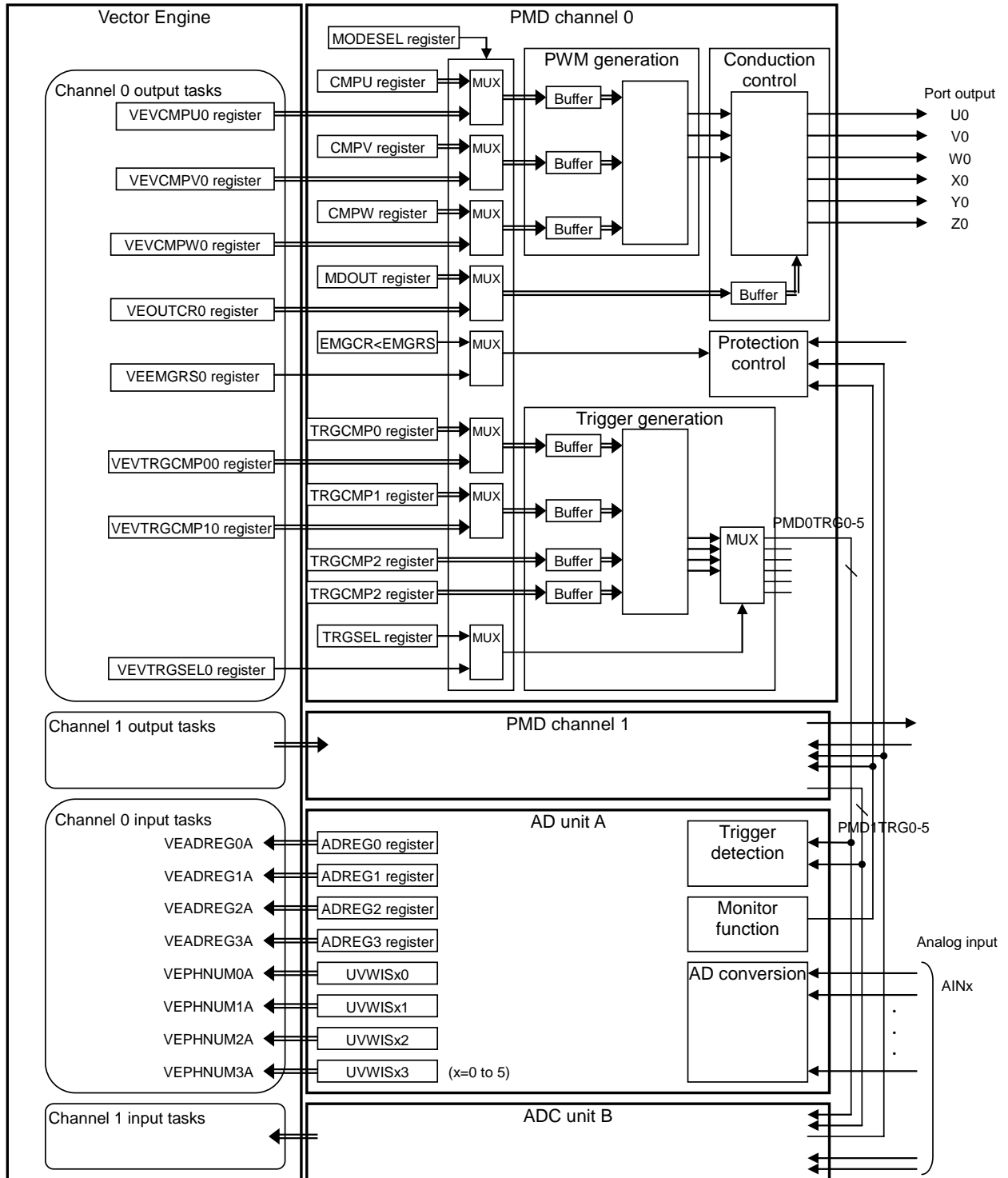


Figure 15-3 Interaction among Vector Engine, PMD and ADC

### 15.3 List of Registers

The Vector Engine registers are divided into the following three types:

VE control registers : Vector Engine control registers and temporary registers

Common registers : Registers common to both channels

Channel-specific registers : Computation data and control registers for each channel

Table 15-1 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          |

Table 15-2 List of Registers (2) – Common Registers

| Address     | Register Name | Channel | Description                              | Read/Write |
|-------------|---------------|---------|--|------------|
| 0x4005-0174 | -             | -       | Reserved                                 | R/W        |
| 0x4005-0178 | VETADC        | Common  | ADC conversion time (based on PWM clock) | R/W        |



Table 15-3 List of Registers (3) – Channel-Specific Registers for Channel 0

| Address     | Register Name | Channel | Description   | Read/Write |
|-------------|---------------|---------|---|------------|
| 0x4005-0044 | VEMCTLF0      | 0       | Status flags  | R/W        |
| 0x4005-0048 | VEMODE0       | 0       | Task control mode   | R/W        |
| 0x4005-004C | VEFMODE0      | 0       | Flow control  | R/W        |
| 0x4005-0050 | VETPWM0       | 0       | PWM period rate (PWM period [s] × maximum speed <sup>*1</sup> × 2 <sup>16</sup> )       | R/W        |
| 0x4005-0054 | VEOMEGA0      | 0       | Rotation speed (speed [Hz] ÷ maximum speed <sup>*1</sup> × 2 <sup>15</sup> )            | R/W        |
| 0x4005-0058 | VETHETA0      | 0       | Motor phase (motor phase [deg]/360 × 2 <sup>16</sup> )                                  | R/W        |
| 0x4005-005C | VEIDREF0      | 0       | d-axis reference value (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>15</sup> ) | R/W        |
| 0x4005-0060 | VEIQREF0      | 0       | q-axis reference value (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>15</sup> ) | R/W        |
| 0x4005-0064 | VEVD0         | 0       | d-axis voltage (voltage [V] ÷ maximum voltage <sup>*3</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-0068 | VEVQ0         | 0       | q-axis voltage (voltage [V] ÷ maximum voltage <sup>*3</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-006C | VECIDKI0      | 0       | Integral coefficient for PI control of d-axis   | R/W        |
| 0x4005-0070 | VECIDKP0      | 0       | Proportional coefficient for PI control of d-axis                                       | R/W        |
| 0x4005-0074 | VECIQKI0      | 0       | Integral coefficient for PI control of q-axis   | R/W        |
| 0x4005-0078 | VECIQKP0      | 0       | Proportional coefficient for PI control of q-axis                                       | R/W        |
| 0x4005-007C | VEVDIH0       | 0       | Upper 32 bits of integral term (VDI ) of d-axis voltage                                 | R/W        |
| 0x4005-0080 | VEVDILH0      | 0       | Lower 32 bits of integral term (VDI) of d-axis voltage                                  | R/W        |
| 0x4005-0084 | VEVQIH0       | 0       | Upper 32 bits of integral term (VQI) of q-axis voltage                                  | R/W        |
| 0x4005-0088 | VEVQILH0      | 0       | Lower 32 bits of integral term (VQI) of q-axis voltage                                  | R/W        |
| 0x4005-008C | VEFPWMCHG0    | 0       | Switching speed (for 2-phase modulation and shift PWM)                                  | R/W        |
| 0x4005-0090 | VEMDPRD0      | 0       | PWM period (to be set identically with PMD's PWM period)                                | R/W        |
| 0x4005-0094 | VEMINPLS0     | 0       | Minimum pulse width   | R/W        |
| 0x4005-0098 | VETRGCR0      | 0       | Synchronizing trigger correction value  | R/W        |
| 0x4005-009C | -             | -       | Reserved  | R/W        |
| 0x4005-00A0 | VECOS0        | 0       | Cosine value at THETA for output conversion (Q15 data)                                  | R/W        |
| 0x4005-00A4 | VESIN0        | 0       | Sine value at THETA for output conversion (Q15 data)                                    | R/W        |
| 0x4005-00A8 | VECOSM0       | 0       | Previous cosine value for input processing (Q15 data)                                   | R/W        |
| 0x4005-00AC | VESINM0       | 0       | Previous sine value for input processing (Q15 data)                                     | R/W        |
| 0x4005-00B0 | VESECTOR0     | 0       | Sector information (0-11)   | R/W        |
| 0x4005-00B4 | VESECTORM0    | 0       | Previous sector information for input processing (0-11)                                 | R/W        |
| 0x4005-00B8 | VEIA00        | 0       | AD conversion result of a-phase zero-current <sup>*4</sup>                              | R/W        |
| 0x4005-00BC | VEIB00        | 0       | AD conversion result of b-phase zero-current <sup>*4</sup>                              | R/W        |
| 0x4005-00C0 | VEIC00        | 0       | AD conversion result of c-phase zero-current <sup>*4</sup>                              | R/W        |
| 0x4005-00C4 | VEIADC0       | 0       | AD conversion result of a-phase current <sup>*4</sup>                                   | R/W        |
| 0x4005-00C8 | VEIBADC0      | 0       | AD conversion result of b-phase current <sup>*4</sup>                                   | R/W        |
| 0x4005-00CC | VEICADC0      | 0       | AD conversion result of c-phase current <sup>*4</sup>                                   | R/W        |
| 0x4005-00D0 | VEVDC0        | 0       | DC supply voltage (voltage [V] ÷ maximum voltage <sup>*3</sup> × 2 <sup>15</sup> )      | R/W        |
| 0x4005-00D4 | VEID0         | 0       | d-axis current (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-00D8 | VEIQ0         | 0       | q-axis current (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-017C | VECMPU0       | 0       | PMD control: CMPU setting   | R/W        |
| 0x4005-0180 | VECMPV0       | 0       | PMD control: CMPV setting   | R/W        |
| 0x4005-0184 | VECMPW0       | 0       | PMD control: CMPW setting   | R/W        |
| 0x4005-0188 | VEOUTCR0      | 0       | PMD control: Output control (MDOUT)   | R/W        |
| 0x4005-018C | VETRGCMP00    | 0       | PMD control: TRGCMP0 setting  | R/W        |
| 0x4005-0190 | VETRGCMP10    | 0       | PMD control: TRGCMP1 setting  | R/W        |
| 0x4005-0194 | VETRGSEL0     | 0       | PMD control: Trigger selection  | R/W        |
| 0x4005-0198 | VEEMGRS0      | 0       | 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<sup>11</sup>

\*3) Maximum voltage: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of AD converter) × 2<sup>12</sup>

\*4) AD conversion results are stored in the upper 12 bits of each 16-bit register.

Table 15-4 List of Registers (4) – Channel-Specific Registers for Channel 1

| Address     | Register Name | Channel | Description   | Read/Write |
|-------------|---------------|---------|---|------------|
| 0x4005-00DC | VEMCTLF1      | 1       | Status flags  | R/W        |
| 0x4005-00E0 | VEMODE1       | 1       | Task control mode   | R/W        |
| 0x4005-00E4 | VEFMODE1      | 1       | Flow control  | R/W        |
| 0x4005-00E8 | VETPWM1       | 1       | PWM period rate (PWM period [s] × maximum speed <sup>*1</sup> × 2 <sup>16</sup> )       | R/W        |
| 0x4005-00EC | VEOMEGA1      | 1       | Rotation speed (speed [Hz] ÷ maximum speed <sup>*1</sup> × 2 <sup>15</sup> )            | R/W        |
| 0x4005-00F0 | VETHETA1      | 1       | Motor phase (motor phase [deg]/360 × 2 <sup>16</sup> )                                  | R/W        |
| 0x4005-00F4 | VEIDREF1      | 1       | d-axis reference value (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>15</sup> ) | R/W        |
| 0x4005-00F8 | VEIQREF1      | 1       | q-axis reference value (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>15</sup> ) | R/W        |
| 0x4005-00FC | VEVD1         | 1       | d-axis voltage (voltage [V] ÷ maximum voltage <sup>*3</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-0100 | VEVQ1         | 1       | q-axis voltage (voltage [V] ÷ maximum voltage <sup>*3</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-0104 | VECIDK1       | 1       | Integral coefficient for PI control of d-axis   | R/W        |
| 0x4005-0108 | VECIDKP1      | 1       | Proportional coefficient for PI control of d-axis                                       | R/W        |
| 0x4005-010C | VECIQK1       | 1       | Integral coefficient for PI control of q-axis   | R/W        |
| 0x4005-0110 | VECIQKP1      | 1       | Proportional coefficient for PI control of q-axis                                       | R/W        |
| 0x4005-0114 | VEVDIH1       | 1       | Upper 32 bits of integral term (VDI) of d-axis voltage                                  | R/W        |
| 0x4005-0118 | VEVDILH1      | 1       | Lower 32 bits of integral term (VDI) of d-axis voltage                                  | R/W        |
| 0x4005-011C | VEVQIH1       | 1       | Upper 32 bits of integral term (VQI) of q-axis voltage                                  | R/W        |
| 0x4005-0120 | VEVQLH1       | 1       | Lower 32 bits of integral term (VQ_I) of q-axis voltage                                 | R/W        |
| 0x4005-0124 | VEFPWMCHG1    | 1       | Switching speed (for 2-phase modulation and shift PWM)                                  | R/W        |
| 0x4005-0128 | VEMDPRD1      | 1       | PWM period (to be set identically with PMD's PWM period)                                | R/W        |
| 0x4005-012C | VEMINPLS1     | 1       | Minimum pulse width   | R/W        |
| 0x4005-0130 | VETRGCR1      | 1       | Synchronizing trigger correction value  | R/W        |
| 0x4005-0134 | -             | -       | Reserved  | R/W        |
| 0x4005-0138 | VECOS1        | 1       | Cosine value at THETA for output conversion (Q15 data)                                  | R/W        |
| 0x4005-013C | VESIN1        | 1       | Sine value at THETA for output conversion (Q15 data)                                    | R/W        |
| 0x4005-0140 | VECOSM1       | 1       | Previous cosine value for input processing (Q15 data)                                   | R/W        |
| 0x4005-0144 | VESINM1       | 1       | Previous sine value for input processing (Q15 data)                                     | R/W        |
| 0x4005-0148 | VESECTOR1     | 1       | Sector information (0-11)   | R/W        |
| 0x4005-014C | VESECTORM1    | 1       | Previous sector information for input processing (0-11)                                 | R/W        |
| 0x4005-0150 | VEIAO1        | 1       | AD conversion result of a-phase zero-current <sup>*4</sup>                              | R/W        |
| 0x4005-0154 | VEIBO1        | 1       | AD conversion result of b-phase zero-current <sup>*4</sup>                              | R/W        |
| 0x4005-0158 | VEICO1        | 1       | AD conversion result of c-phase zero-current <sup>*4</sup>                              | R/W        |
| 0x4005-015C | VEIAADC1      | 1       | AD conversion result of a-phase current <sup>*4</sup>                                   | R/W        |
| 0x4005-0160 | VEIBADC1      | 1       | AD conversion result of b-phase current <sup>*4</sup>                                   | R/W        |
| 0x4005-0164 | VEICADC1      | 1       | AD conversion result of c-phase current <sup>*4</sup>                                   | R/W        |
| 0x4005-0168 | VEVDC1        | 1       | DC supply voltage (voltage [V] ÷ maximum voltage <sup>*3</sup> × 2 <sup>15</sup> )      | R/W        |
| 0x4005-016C | VEID1         | 1       | d-axis current (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-0170 | VEIQ1         | 1       | q-axis current (current [A] ÷ maximum current <sup>*2</sup> × 2 <sup>31</sup> )         | R/W        |
| 0x4005-019C | VECMPU1       | 1       | PMD control: CMPU setting   | R/W        |
| 0x4005-01A0 | VECMPV1       | 1       | PMD control: CMPV setting   | R/W        |
| 0x4005-01A4 | VECMPW1       | 1       | PMD control: CMPW setting   | R/W        |
| 0x4005-01A8 | VEOUTCR1      | 1       | PMD control: Output control (MDOUT)   | R/W        |
| 0x4005-01AC | VETRGCMP01    | 1       | PMD control: TRGCMP0 setting  | R/W        |
| 0x4005-01B0 | VETRGCMP11    | 1       | PMD control: TRGCMP1 setting  | R/W        |
| 0x4005-01B4 | VETRGSEL1     | 1       | PMD control: Trigger selection  | R/W        |
| 0x4005-01B8 | VEEMGRS1      | 1       | 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<sup>11</sup>

\*3) Maximum voltage: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of AD converter) × 2<sup>12</sup>

\*4) AD conversion results are stored in the upper 12 bits of each 16-bit register.

## 15.4 Description of Registers

### 15.4.1 VE Control Registers

#### 15.4.1.1 VEEN Register

|                       |             |     |  |  |   |  |   |                                      |
|-----------------------|-------------|-----|--|--|---|--|---|--------------------------------------|
|                       | 31          | ... |  |  |   | 9  | 8 |                                      |
| VEEN<br>(0x4005_0000) | bit Symbol  |     |  |  |   |  |   |                                      |
|                       | Read/Write  |     |  |  |   |  |   |                                      |
|                       | After reset |     |  |  |   |  |   |                                      |
|                       | Function    |     |  |  |   |  |   |                                      |
|                       | 7           | ... |  |  | 3 | 2  | 1 | 0                                    |
|                       | bit Symbol  |     |  |  |   | VEIDLEN  |   | VEEN                                 |
|                       | Read/Write  |     |  |  |   | R/W  |   | R/W                                  |
|                       | After reset |     |  |  |   | 0  |   | 0                                    |
|                       | Function    |     |  |  |   | Operation in IDLE mode<br>0: Inactive<br>1: Active |   | VE enable<br>0: Disable<br>1: Enable |

<VEEN> Disables or enables the Vector Engine.

<VEIDLEN> Controls whether or not the clock is supplied to the Vector Engine in IDLE mode.

#### 15.4.1.2 VECPURUNTRG Register

|                              |             |     |  |  |   |   |   |   |
|------------------------------|-------------|-----|--|--|---|---|---|---|
|                              | 31          | ... |  |  |   | 9   | 8 |   |
| VECPURUNTRG<br>(0x4005_0004) | bit Symbol  |     |  |  |   |   |   |   |
|                              | Read/Write  |     |  |  |   |   |   |   |
|                              | After reset |     |  |  |   |   |   |   |
|                              | Function    |     |  |  |   |   |   |   |
|                              | 7           | ... |  |  | 3 | 2   | 1 | 0   |
|                              | bit Symbol  |     |  |  |   | VCPURTB                                     |   | VCPURTA                                     |
|                              | Read/Write  |     |  |  |   | W   |   | W   |
|                              | After reset |     |  |  |   | 0   |   | 0   |
|                              | Function    |     |  |  |   | Channel 1 start command<br>0: -<br>1: Start |   | Channel 0 start command<br>0: -<br>1: Start |

<VCPURTA> Starts channel 0 by programming.

<VCPURTB> Starts channel 1 by programming.

- \* 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.

15.4.1.3 VETASKAPP Register

|                            |  |     |   |   |  |   |   |   |
|----------------------------|--|-----|---|---|--|---|---|---|
|                            | 31                                     | ... |   |   |  | 9 | 8 |   |
| VETASKAPP<br>(0x4005_0008) | bit Symbol                             |     |   |   |  |   |   |   |
|                            | Read/Write                             |     |   |   |  |   |   |   |
|                            | After reset                            |     |   |   |  |   |   |   |
|                            | Function                               |     |   |   |  |   |   |   |
|                            | 7                                      | 6   | 5 | 4 | 3                                      | 2 | 1 | 0 |
|                            | bit Symbol                             |     |   |   | VTASKA                                 |   |   |   |
|                            | Read/Write                             |     |   |   | VTASKB                                 |   |   |   |
|                            | After reset                            |     |   |   | R/W                                    |   |   |   |
|                            | Function                               |     |   |   | R/W                                    |   |   |   |
|                            | 0x0                                    |     |   |   | 0x0                                    |   |   |   |
|                            | Channel 1 task selection               |     |   |   | Channel 0 task selection               |   |   |   |
|                            | 0x0: Output control                    |     |   |   | 0x0: Output control                    |   |   |   |
|                            | 0x1: Trigger generation                |     |   |   | 0x1: Trigger generation                |   |   |   |
|                            | 0x2: Input processing                  |     |   |   | 0x2: Input processing                  |   |   |   |
|                            | 0x3: Input phase conversion            |     |   |   | 0x3: Input phase conversion            |   |   |   |
|                            | 0x4: Input coordinate axis conversion  |     |   |   | 0x4: Input coordinate axis conversion  |   |   |   |
|                            | 0x5: Current control                   |     |   |   | 0x5: Current control                   |   |   |   |
|                            | 0x6: SIN/COS computation               |     |   |   | 0x6: SIN/COS computation               |   |   |   |
|                            | 0x7: Output coordinate axis conversion |     |   |   | 0x7: Output coordinate axis conversion |   |   |   |
|                            | 0x8: Output phase conversion           |     |   |   | 0x8: Output phase conversion           |   |   |   |
|                            | 0x9 -0xF: Reserved                     |     |   |   | 0x9 -0xF: Reserved                     |   |   |   |

<VTASKA> Specifies the task to be performed when channel 0 is started by programming.  
 <VTASKB> Specifies the task to be performed when channel 1 is started by programming.

\* Only those tasks that are included in schedules can be specified.

15.4.1.4 VEACTION Register

|                           |                                |     |   |   |                                |   |   |   |
|---------------------------|--------------------------------|-----|---|---|--------------------------------|---|---|---|
|                           | 31                             | ... |   |   |                                | 9 | 8 |   |
| VEACTION<br>(0x4005_000C) | bit Symbol                     |     |   |   |                                |   |   |   |
|                           | Read/Write                     |     |   |   |                                |   |   |   |
|                           | After reset                    |     |   |   |                                |   |   |   |
|                           | Function                       |     |   |   |                                |   |   |   |
|                           | 7                              | 6   | 5 | 4 | 3                              | 2 | 1 | 0 |
|                           | bit Symbol                     |     |   |   | VACTA                          |   |   |   |
|                           | Read/Write                     |     |   |   | VACTB                          |   |   |   |
|                           | After reset                    |     |   |   | R/W                            |   |   |   |
|                           | Function                       |     |   |   | R/W                            |   |   |   |
|                           | 0x0                            |     |   |   | 0x0                            |   |   |   |
|                           | Channel 1 schedule             |     |   |   | Channel 0 schedule             |   |   |   |
|                           | 0x0: Individual task execution |     |   |   | 0x0: Individual task execution |   |   |   |
|                           | 0x1: Schedule 1                |     |   |   | 0x1: Schedule 1                |   |   |   |
|                           | 0x4: Schedule 4                |     |   |   | 0x4: Schedule 4                |   |   |   |
|                           | 0x9: Schedule 9                |     |   |   | 0x9: Schedule 9                |   |   |   |
|                           | Other: Reserved                |     |   |   | Other: Reserved                |   |   |   |

<VACTA> Specifies an individual task execution or a schedule for channel 0.  
 <VACTB> Specifies an individual task execution or a schedule for channel 1.

15.4.1.5 VEREPTIME Register

|                            |  |          |   |   |  |   |   |   |
|----------------------------|--|----------|---|---|--|---|---|---|
| VEREPTIME<br>(0x4005_0010) | 31   | ...      |   |   |  | 9 | 8 |   |
|                            | bit Symbol   |          |   |   |  |   |   |   |
|                            | Read/Write   | R        |   |   |  |   |   |   |
|                            | After reset  | 0x000000 |   |   |  |   |   |   |
|                            | Function   | -        |   |   |  |   |   |   |
|                            | 7  | 6        | 5 | 4 | 3  | 2 | 1 | 0 |
| bit Symbol                 | VREPB  |          |   |   | VREPA  |   |   |   |
| Read/Write                 | R/W  |          |   |   | R/W  |   |   |   |
| After reset                | 0x0  |          |   |   | 0x0  |   |   |   |
| Function                   | Channel 1 repeat count<br>0: Do not execute schedule<br>1-15: Execute schedule a specified number of times |          |   |   | Channel 0 repeat count<br>0: Do not execute schedule<br>1-15: Execute schedule a specified number of times |   |   |   |

<VREPA> Specifies the repeat times a schedule is to be executed in channel 0.

<VREPB> Specifies the repeat times a schedule is to be executed in channel 1.

\* When "0" is set, no schedule is executed.

15.4.1.6 VETRGMODE Register

|                            |             |          |   |   |  |   |  |   |
|----------------------------|-------------|----------|---|---|--|---|--|---|
| VETRGMODE<br>(0x4005_0014) | 31          | ...      |   |   |  | 9 | 8  |   |
|                            | bit Symbol  |          |   |   |  |   |  |   |
|                            | Read/Write  | R        |   |   |  |   |  |   |
|                            | After reset | 0x000000 |   |   |  |   |  |   |
|                            | Function    | -        |   |   |  |   |  |   |
|                            | 7           | 6        | 5 | 4 | 3  | 2 | 1  | 0 |
| bit Symbol                 |             |          |   |   | VTRGB  |   | VTRGA  |   |
| Read/Write                 | R           |          |   |   | R/W  |   | R/W  |   |
| After reset                | 0x0         |          |   |   | 00   |   | 00   |   |
| Function                   | -           |          |   |   | Channel 1 trigger mode<br>00: Ignore both INTB0 (unit A) and INTB1 (unit B)<br>01: Start by INTB0 (unit A)<br>10: Start by INTB1 (unit B)<br>11: Start when both INTB0 (unit A) and INTB1 (unit B) occur |   | Channel 0 trigger mode<br>00: Ignore both INTA0 (unit A) and INTA1 (unit B)<br>01: Start by INTA0 (unit A)<br>10: Start by INTA1 (unit B)<br>11: Start when both INTA0 (unit A) and INTA1 (unit B) occur |   |

<VTRGA> Specifies the AD conversion end interrupt that triggers input processing in channel 0.

<VTRGB> Specifies the AD conversion end interrupt that triggers input processing in channel 1.

15.4.1.7 VEERRINTEN Register

|                             |             |          |  |   |  |  |   |
|-----------------------------|-------------|----------|--|---|--|--|---|
|                             | 31          | ...      |  |   |  | 9  | 8 |
| VEERRINTEN<br>(0x4005_0018) | bit Symbol  |          |  |   |  |  |   |
|                             | Read/Write  | R        |  |   |  |  |   |
|                             | After reset | 0x000000 |  |   |  |  |   |
|                             | Function    | -        |  |   |  |  |   |
|                             | 7           | ...      |  | 3 | 2  | 1  | 0 |
| bit Symbol                  |             |          |  |   | VERRENB  | VERRENA  |   |
| Read/Write                  | R           |          |  |   | R/W  | R/W  |   |
| After reset                 | 000000      |          |  |   | 0  | 0  |   |
| Function                    | -           |          |  |   | Channel 1 error<br>interrupt enable<br>0: Disable<br>1: Enable | Channel 0 error<br>interrupt enable<br>0: Disable<br>1: Enable |   |

<VERRENA> Enables or disables the error detection interrupt in channel 0.  
 <VERRENB> Enables or disables the error detection interrupt in channel 1.

15.4.1.8 VECOMPEND Register

|                            |             |          |  |   |   |   |   |
|----------------------------|-------------|----------|--|---|---|---|---|
|                            | 31          | ...      |  |   |   | 9   | 8 |
| VECOMPEND<br>(0x4005_001C) | bit Symbol  |          |  |   |   |   |   |
|                            | Read/Write  | R        |  |   |   |   |   |
|                            | After reset | 0x000000 |  |   |   |   |   |
|                            | Function    | -        |  |   |   |   |   |
|                            | 7           | ...      |  | 3 | 2   | 1   | 0 |
| bit Symbol                 |             |          |  |   | VCENDB  | VCENDA  |   |
| Read/Write                 | R           |          |  |   | W   | W   |   |
| After reset                | 000000      |          |  |   | 0   | 0   |   |
| Function                   | -           |          |  |   | Channel 1 forced<br>termination<br>0: -<br>1: Terminate | Channel 0 forced<br>termination<br>0:<br>1: Terminate |   |

<VCENDA> Forcefully terminates the currently executing schedule in channel 0.  
 <VCENDB> Forcefully terminates the currently executing schedule in channel 1.  
 \* When "1" is written to these bits, it is cleared in the next cycle. These bits always read as "0".

15.4.1.9 VEERRDET Register

|                           |             |     |  |   |   |   |   |
|---------------------------|-------------|-----|--|---|---|---|---|
|                           | 31          | ... |  |   |   | 9 | 8   |
| VEERRDET<br>(0x4005_0020) | bit Symbol  |     |  |   |   |   |   |
|                           | Read/Write  |     |  |   |   |   |   |
|                           | After reset |     |  |   |   |   |   |
|                           | Function    |     |  |   |   |   |   |
|                           | 7           | ... |  | 3 | 2   | 1 | 0   |
|                           | bit Symbol  |     |  |   | VERRDB  |   | VERRDA  |
|                           | Read/Write  |     |  |   | R   |   | R   |
|                           | After reset |     |  |   | 0   |   | 0   |
|                           | Function    |     |  |   | Channel 1 error flag<br>0: No error detected<br>1: Error detected |   | Channel 0 error flag<br>0: No error detected<br>1: Error detected |

<VERRDA> Channel 0 error flag

<VERRDB> Channel 1 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.

15.4.1.10 VESCHTASKRUN Register

|                               |   |     |   |   |   |   |    |   |   |   |
|-------------------------------|---|-----|---|---|---|---|----|---|---|---|
|                               | 31  | ... |   |   |   | 17  | 16 |   |   |   |
| VESCHTASKRUN<br>(0x4005_0024) | bit Symbol  |     |   |   |   |   |    |   |   |   |
|                               | Read/Write  |     |   |   |   |   |    |   |   |   |
|                               | After reset   |     |   |   |   |   |    |   |   |   |
|                               | Function  |     |   |   |   |   |    |   |   |   |
|                               | 15  | ... |   |   |   | 11  | 10 |   |   |   |
|                               | bit Symbol  |     |   |   |   |   |    |   |   |   |
|                               | Read/Write  |     |   |   |   |   |    |   |   |   |
|                               | After reset   |     |   |   |   |   |    |   |   |   |
|                               | Function  |     |   |   |   |   |    |   |   |   |
|                               | 9   | 8   | 7 | 6 | 5   | 4   | 3  | 2 | 1 | 0   |
|                               | VRTASKB   |     |   |   | VRSCHB  | VRTASKA   |    |   |   | VRSCHA  |
|                               | R   |     |   |   | R   | R   |    |   |   | R   |
|                               | 0x0   |     |   |   | 0   | 0x0   |    |   |   | 0   |
|                               | Task executing in channel 1<br>0x0: Output control<br>0x1: Trigger generation<br>0x2: Input processing<br>0x3: Input phase conversion<br>0x4: Input coordinate axis conversion<br>0x5: Current control<br>0x6: SIN/COS computation<br>0x7: Output coordinate axis conversion<br>0x8: Output phase conversion<br>0x9-0xF: Reserved |     |   |   | Channel 1 schedule status<br>0: Not executing<br>1: Executing | Task executing in channel 0<br>0x0: Output control<br>0x1: Trigger generation<br>0x2: Input processing<br>0x3: Input phase conversion<br>0x4: Input coordinate axis conversion<br>0x5: Current control<br>0x6: SIN/COS computation<br>0x7: Output coordinate axis conversion<br>0x8: Output phase conversion<br>0x9-0xF: Reserved |    |   |   | Channel 0 schedule status<br>0: Not executing<br>1: Executing |

<VRSCHA> Schedule execution status in channel 0

<VRTASKA> Task Number currently executing in channel 0

<VRSCHB> Schedule execution status in channel 1

<VRTASKB> Task Number currently executing in channel 1

## 15.4.1.11 Temporary Registers

|                            |             |            |     |   |   |
|----------------------------|-------------|------------|-----|---|---|
|                            |             | 31         | ... | 1 | 0 |
| VETMPREG0<br>(0x4005_002C) | bit Symbol  | TMPREG0    |     |   |   |
|                            | Read/Write  | R/W        |     |   |   |
|                            | After reset | 0x00000000 |     |   |   |
|                            | Function    |            |     |   |   |
|                            |             | 31         | ... | 1 | 0 |
| VETMPREG1<br>(0x4005_0030) | bit Symbol  | TMPREG1    |     |   |   |
|                            | Read/Write  | R/W        |     |   |   |
|                            | After reset | 0x00000000 |     |   |   |
|                            | Function    |            |     |   |   |
|                            |             | 31         | ... | 1 | 0 |
| VETMPREG2<br>(0x4005_0034) | bit Symbol  | TMPREG2    |     |   |   |
|                            | Read/Write  | R/W        |     |   |   |
|                            | After reset | 0x00000000 |     |   |   |
|                            | Function    |            |     |   |   |
|                            |             | 31         | ... | 1 | 0 |
| VETMPREG3<br>(0x4005_0038) | bit Symbol  | TMPREG3    |     |   |   |
|                            | Read/Write  | R/W        |     |   |   |
|                            | After reset | 0x00000000 |     |   |   |
|                            | Function    |            |     |   |   |
|                            |             | 31         | ... | 1 | 0 |
| VETMPREG4<br>(0x4005_003C) | bit Symbol  | TMPREG4    |     |   |   |
|                            | Read/Write  | R/W        |     |   |   |
|                            | After reset | 0x00000000 |     |   |   |
|                            | Function    |            |     |   |   |
|                            |             | 31         | ... | 1 | 0 |
| VETMPREG5<br>(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



15.4.2 Common Registers

15.4.2.1 VETADC Register

|                         |                                      |        |     |    |    |
|-------------------------|--------------------------------------|--------|-----|----|----|
| VETADC<br>(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<br>0x0000-0xFFFF |        |     |    |    |

<TADC> The value to be set is:

$$\text{ADC conversion time [s]} \div \text{PWM counter clock frequency [s]}$$

- \* This register is effective when the 1-shunt current detection mode is selected and PWM shift is enabled.

### 15.4.3 Channel-Specific Registers

#### 15.4.3.1 VEMODE Register

|                          |             |              |   |   |   |   |  |   |   |
|--------------------------|-------------|--------------|---|---|---|---|--|---|---|
|                          | 31          | ...          |   |   |   | 9 | 8  |   |   |
| VEMODE0<br>(0x4005_0048) | bit Symbol  |              |   |   |   |   |  |   |   |
|                          | Read/Write  | R            |   |   |   |   |  |   |   |
| VEMODE1<br>(0x4005_00E0) | 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<br>00: Output OFF<br>01: Output enable<br>10: Reserved<br>11: Output OFF and<br>EMG return |   | Zero-current<br>detection<br>0: Disable<br>1: Enable | Phase<br>interpolation<br>0: Disable<br>1: Enable |   |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

- <PVIEN> Enables or disables phase interpolation.
- <ZIEN> Enables or disables zero-current detection.
- <OCRMD> Controls output operation (output OFF, output enable or EMG return).

15.4.3.2 VEFMODE Register

|                           |              |     |  |    |              |   |   |   |   |   |
|---------------------------|--------------|-----|--|----|--------------|---|---|---|---|---|
|                           | 31           | ... |  |    |              |   | 17  | 16  |   |   |
| VEFMODE0<br>(0x4005_004C) | bit Symbol   |     |  |    |              |   |   |   |   |   |
|                           | Read/Write   |     |  |    |              |   |   |   |   |   |
|                           | R            |     |  |    |              |   |   |   |   |   |
| VEFMODE1<br>(0x4005_00E4) | 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   | 0   |   |   |   |
|                           | Function     |     |  |    |              |   | Keep the previous value of SIN/COS/SECTOR | Write as '0'  |   |   |
|                           | Write as '0' |     |  |    |              |   | 0: effective                              | 1: no_effective   |   |   |
|                           | 7            | 6   | 5  | 4  | 3            | 2   | 1   | 0   |   |   |
|                           | bit Symbol   |     | ADCSEL   |    | -            | PMDSEL                                      |   | IDMODE  | SPWMEN                                      | C2PEN   |
|                           | Read/Write   |     | R/W  |    | R/W          | R/W   |   | R/W   | R/W   | R/W   |
|                           | After reset  |     | 00   |    | 0            | 00  |   | 0   | 0   | 0   |
|                           | Function     |     | ADC unit<br>00: Unit A<br>01: Unit B<br>10: Unit A,B<br>11: Unit A,B |    | Write as '0' | PMD channel<br>0: Channel 0<br>1: Channel 1 |   | Current detection mode<br>00: 3-shunt<br>01: 2-sensor<br>10: 1-shunt (for up count PMDTRG)<br>11: 1-shunt (for down count PMDTRG) | PWM shift enable<br>0: Disable<br>1: Enable | Modulation mode<br>0: 3-phase modulation<br>1: 2-phase modulation |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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.

| VEFMODE0<br><IDMODE> | VEFMODE1<br><IDMODE> | PMD0TRGCR<br><TRG0MD> | PMD0TRGCR<br><TRG1MD> | PMD1TRGCR<br><TRG0MD> | PMD1TRGCR<br><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)  | -                     | -                     |
| -                    | 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)  |

<PMDSEL>: Selects PMD channel 0 or channel 1.

\* Select PMD channel 0 when Vector Engine channel 0 is used .

Select PMD channel 1 when Vector Engine channel 1 is used .

<ADCSEL>: Selects the ADC unit to be used.

\* Select unit A or unit A/unit B when Vector Engine channel 0 is used .

Select unit B or unit A/unit B when Vector Engine channel 1 is used

< MREGDIS > : Keep or Not\_keep the previous value of SIN/COS/SECTOR.

\* In case of no\_effective, SINM=SIN,COSM=COS,SECTORM=SECTOR.

15.4.3.3 VETPWM Register

|                          |             |                                     |    |    |
|--------------------------|-------------|-------------------------------------|----|----|
|                          | 31          | ...                                 | 17 | 16 |
| VETPWM0<br>(0x4005_0050) | bit Symbol  |                                     |    |    |
|                          | Read/Write  | R                                   |    |    |
| VETPWM1<br>(0x4005_00E8) | After reset | 0x0000                              |    |    |
|                          | Function    | -                                   |    |    |
|                          | 15          | ...                                 | 1  | 0  |
|                          | bit Symbol  | TPWM                                |    |    |
|                          | Read/Write  | R/W                                 |    |    |
|                          | After reset | 0x0000                              |    |    |
|                          | Function    | PWM period rate<br>0x0000 to 0xFFFF |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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:  
 $PWM\ period\ [s] \times Max\_Hz \times 2^{16}$   
 \* Max\_Hz: Maximum rotation speed  
 \* It indicates a ratio between PWM frequency and maximum rotation speed.

15.4.3.4 VEOMEGA Register

|                           |             |                                    |    |    |
|---------------------------|-------------|------------------------------------|----|----|
|                           | 31          | ...                                | 17 | 16 |
| VEOMEGA0<br>(0x4005_0054) | bit Symbol  |                                    |    |    |
|                           | Read/Write  | R                                  |    |    |
| VEOMEGA1<br>(0x4005_00EC) | After reset | 0x0000                             |    |    |
|                           | Function    | -                                  |    |    |
|                           | 15          | ...                                | 1  | 0  |
|                           | bit Symbol  | OMEGA                              |    |    |
|                           | Read/Write  | R/W                                |    |    |
|                           | After reset | 0x0000                             |    |    |
|                           | Function    | Rotation speed<br>0x0000 to 0xFFFF |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<OMEGA> Set a rotation speed (16-bit fixed-point data: -1.0 to 1.0) as follows:  
 $Rotation\ speed\ [Hz] \div Max\_Hz \times 2^{15}$   
 \* Max\_Hz: Maximum rotation speed [Hz]

15.4.3.5 VETHETA Register

|                           |             |            |    |    |
|---------------------------|-------------|------------|----|----|
|                           | 31          | ...        | 17 | 16 |
| VETHETA0<br>(0x4005_0058) | bit Symbol  |            |    |    |
|                           | Read/Write  | R          |    |    |
| VETHETA1<br>(0x4005_00F0) | After reset | 0x0000     |    |    |
|                           | Function    | -          |    |    |
|                           | 15          | ...        | 1  | 0  |
|                           | bit Symbol  | THETA      |    |    |
|                           | Read/Write  | R/W        |    |    |
|                           | After reset | 0x0000     |    |    |
|                           | Function    | Phase data |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<THETA> Set phase data (16-bit fixed-point data: 0.0 to 1.0) as follows:  
 $Phase\ [deg] \div 360 \times 2^{16}$

15.4.3.6 VESIN/COS Registers

|                          |  |        |     |    |    |
|--------------------------|--|--------|-----|----|----|
|                          |  | 31     | ... | 17 | 16 |
| VECOS0<br>(0x4005_00A0)  | bit Symbol                                   |        |     |    |    |
|                          | Read/Write                                   | R      |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
|                          | Function                                     | -      |     |    |    |
| VECOS1<br>(0x4005_0138)  |  | 15     | ... | 1  | 0  |
|                          | bit Symbol                                   | COS    |     |    |    |
|                          | Read/Write                                   | R/W    |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
| Function                 | Cosine value: 0x0000-0xFFFF                  |        |     |    |    |
|                          |  | 31     | ... | 17 | 16 |
| VESIN0<br>(0x4005_00A4)  | bit Symbol                                   |        |     |    |    |
|                          | Read/Write                                   | R      |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
|                          | Function                                     | -      |     |    |    |
| VESIN1<br>(0x4005_013C)  |  | 15     | ... | 1  | 0  |
|                          | bit Symbol                                   | SIN    |     |    |    |
|                          | Read/Write                                   | R/W    |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
| Function                 | Sine value: 0x0000-0xFFFF                    |        |     |    |    |
|                          |  | 31     | ... | 17 | 16 |
| VECOSM0<br>(0x4005_00A8) | bit Symbol                                   |        |     |    |    |
|                          | Read/Write                                   | R      |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
|                          | Function                                     | -      |     |    |    |
| VECOSM1<br>(0x4005_0140) |  | 15     | ... | 1  | 0  |
|                          | bit Symbol                                   | COSM   |     |    |    |
|                          | Read/Write                                   | R/W    |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
| Function                 | Cosine value (previous value): 0x0000-0xFFFF |        |     |    |    |
|                          |  | 31     | ... | 17 | 16 |
| VESINM0<br>(0x4005_00AC) | bit Symbol                                   |        |     |    |    |
|                          | Read/Write                                   | R      |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
|                          | Function                                     | -      |     |    |    |
| VESINM1<br>(0x4005_0144) |  | 15     | ... | 1  | 0  |
|                          | bit Symbol                                   | SINM   |     |    |    |
|                          | Read/Write                                   | R/W    |     |    |    |
|                          | After reset                                  | 0x0000 |     |    |    |
| Function                 | Sine value (previous value): 0x0000-0xFFFF   |        |     |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

- <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

15.4.3.7 dq Current Reference Registers

|                           |             |   |    |    |
|---------------------------|-------------|---|----|----|
|                           | 31          | ...   | 17 | 16 |
| VEIDREF0<br>(0x4005_005C) | bit Symbol  | IDREF   |    |    |
|                           | Read/Write  | R   |    |    |
| VEIDREF1<br>(0x4005_00F4) | After reset | 0x0000  |    |    |
|                           | Function    | -   |    |    |
|                           | 15          | ...   | 1  | 0  |
| VEIQREF0<br>(0x4005_0060) | bit Symbol  | IQREF   |    |    |
|                           | Read/Write  | R/W   |    |    |
| VEIQREF1<br>(0x4005_00F8) | 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 |
| VEIQREF0<br>(0x4005_0060) | bit Symbol  | IQREF   |    |    |
|                           | Read/Write  | R   |    |    |
| VEIQREF1<br>(0x4005_00F8) | 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 |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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:

$$\text{Axis current reference [A]} \div \text{Max}_I \times 2^{15}$$

\* Max<sub>I</sub>: (Phase current value [A] which corresponds to 1 LSB of ADC) × 2<sup>11</sup>

15.4.3.8 dq Voltage Registers

|                        |             |                                       |   |   |
|------------------------|-------------|---------------------------------------|---|---|
|                        | 31          | ...                                   | 1 | 0 |
| VEVD0<br>(0x4005_0064) | bit Symbol  | VD                                    |   |   |
|                        | Read/Write  | R/W                                   |   |   |
| VEVD1<br>(0x4005_00FC) | After reset | 0x00000000                            |   |   |
|                        | Function    | d-axis voltage: 0x00000000-0xFFFFFFFF |   |   |
|                        | 31          | ...                                   | 1 | 0 |
| VEVQ0<br>(0x4005_0068) | bit Symbol  | VQ                                    |   |   |
|                        | Read/Write  | R/W                                   |   |   |
| VEVQ1<br>(0x4005_0100) | After reset | 0x00000000                            |   |   |
|                        | Function    | q-axis voltage: 0x00000000-0xFFFFFFFF |   |   |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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:

$$\text{Axis voltage} \div \text{Max}_V \times 2^{31}$$

\* Max<sub>V</sub>: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC) × 2<sup>12</sup>

15.4.3.9 PI Control Coefficient Registers

|                           |             |  |     |    |    |
|---------------------------|-------------|--|-----|----|----|
|                           |             | 31   | ... | 17 | 16 |
| VECIDKI0<br>(0x4005_006C) | bit Symbol  |  |     |    |    |
|                           | Read/Write  | R  |     |    |    |
| VECIDKI1<br>(0x4005_0104) | 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     |     |    |    |
|                           |             | 31   | ... | 17 | 16 |
| VECIDKP0<br>(0x4005_0070) | bit Symbol  |  |     |    |    |
|                           | Read/Write  | R  |     |    |    |
| VECIDKP1<br>(0x4005_0108) | 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 |     |    |    |
|                           |             | 31   | ... | 17 | 16 |
| VECIQKI0<br>(0x4005_0074) | bit Symbol  |  |     |    |    |
|                           | Read/Write  | R  |     |    |    |
| VECIQKI1<br>(0x4005_010C) | 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     |     |    |    |
|                           |             | 31   | ... | 17 | 16 |
| VECIQKP0<br>(0x4005_0078) | bit Symbol  |  |     |    |    |
|                           | Read/Write  | R  |     |    |    |
| VECIQKP1<br>(0x4005_0110) | 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 |     |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

- <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

15.4.3.10 PI Control Integral Term Registers

|                          |             |                      |   |   |
|--------------------------|-------------|----------------------|---|---|
|                          | 31          | ...                  | 1 | 0 |
| VEVDIH0<br>(0x4005_007C) | bit Symbol  | VDIH                 |   |   |
|                          | Read/Write  | R/W                  |   |   |
| VEVD_H1<br>(0x4005_0114) | After reset | 0x00000000           |   |   |
|                          | Function    | Upper 32 bits of VDI |   |   |

|                           |             |        |    |    |
|---------------------------|-------------|--------|----|----|
|                           | 31          | ...    | 17 | 16 |
| VEVDILH0<br>(0x4005_0080) | bit Symbol  | VDILH  |    |    |
|                           | Read/Write  | R/W    |    |    |
| VEVDILH1<br>(0x4005_0118) | After reset | 0x0000 |    |    |
|                           | Function    |        |    |    |

|  |             |        |   |   |
|--|-------------|--------|---|---|
|  | 15          | ...    | 1 | 0 |
|  | bit Symbol  |        |   |   |
|  | Read/Write  | R      |   |   |
|  | After reset | 0x0000 |   |   |
|  | Function    | -      |   |   |

|                          |             |            |   |   |
|--------------------------|-------------|------------|---|---|
|                          | 31          | ...        | 1 | 0 |
| VEVQIH0<br>(0x4005_0084) | bit Symbol  | VQIH       |   |   |
|                          | Read/Write  | R/W        |   |   |
| VEVQIH1<br>(0x4005_011C) | After reset | 0x00000000 |   |   |
|                          | Function    |            |   |   |

|                           |             |        |    |    |
|---------------------------|-------------|--------|----|----|
|                           | 31          | ...    | 17 | 16 |
| VEVQILH0<br>(0x4005_0088) | bit Symbol  | VQILH  |    |    |
|                           | Read/Write  | R/W    |    |    |
| VEVQILH1<br>(0x4005_0120) | After reset | 0x0000 |    |    |
|                           | Function    |        |    |    |

|  |             |        |   |   |
|--|-------------|--------|---|---|
|  | 15          | ...    | 1 | 0 |
|  | bit Symbol  |        |   |   |
|  | Read/Write  | R      |   |   |
|  | After reset | 0x0000 |   |   |
|  | Function    | -      |   |   |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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)





15.4.3.13 PWM Period Register

|                            |             |            |    |    |
|----------------------------|-------------|------------|----|----|
|                            | 31          | ...        | 17 | 16 |
| VEVMDPRD0<br>(0x4005_0090) | bit Symbol  |            |    |    |
|                            | Read/Write  | R          |    |    |
| VEVMDPRD1<br>(0x4005_0128) | After reset | 0x0000     |    |    |
|                            | Function    | -          |    |    |
|                            | 15          | ...        | 1  | 0  |
|                            | bit Symbol  | VMDPRD     |    |    |
|                            | Read/Write  | R/W        |    |    |
|                            | After reset | 0x0000     |    |    |
|                            | Function    | PWM period |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<VMDPRD> PWM period. Set the value of the PMD's MDPRD register.

15.4.3.14 VEMINPLS Register

|                            |             |                                  |    |    |
|----------------------------|-------------|----------------------------------|----|----|
|                            | 31          | ...                              | 17 | 16 |
| VEMINPLS0<br>(0x4005_0094) | bit Symbol  |                                  |    |    |
|                            | Read/Write  | R                                |    |    |
| VEMINPLS1<br>(0x4005_012C) | After reset | 0x0000                           |    |    |
|                            | Function    | -                                |    |    |
|                            | 15          | ...                              | 1  | 0  |
|                            | bit Symbol  | MINPLS                           |    |    |
|                            | Read/Write  | R/W                              |    |    |
|                            | After reset | 0x0000                           |    |    |
|                            | Function    | Minimum disparity of pulse width |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<MINPLS> Set the minimum disparity of pulse width among the duty of VECMPU, VECMPV, VECMPW.

$$\text{Disparity of pulse width [s]} \div \text{PWM counter clock period [s]}$$

15.4.3.15 Sector Registers

|                            |             |          |     |   |   |                                |   |   |   |
|----------------------------|-------------|----------|-----|---|---|--------------------------------|---|---|---|
|                            |             | 31       | ... |   |   |                                | 9 | 8 |   |
| VESECTOR0<br>(0x4005_00B0) | bit Symbol  |          |     |   |   |                                |   |   |   |
|                            | Read/Write  | R        |     |   |   |                                |   |   |   |
| VESECTOR1<br>(0x4005_0148) | 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<br>0x0-0xF: |   |   |   |

|                             |             |          |     |   |   |                                |   |   |   |
|-----------------------------|-------------|----------|-----|---|---|--------------------------------|---|---|---|
|                             |             | 31       | ... |   |   |                                | 9 | 8 |   |
| VESECTORM0<br>(0x4005_00B4) | bit Symbol  |          |     |   |   |                                |   |   |   |
|                             | Read/Write  | R        |     |   |   |                                |   |   |   |
| VESECTORM1<br>(0x4005_014C) | 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<br>0x0-0xF: |   |   |   |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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.

15.4.3.16 Zero-Current Registers

|                         |             |     |    |    |
|-------------------------|-------------|-----|----|----|
|                         | 31          | ... | 17 | 16 |
| VEIA00<br>(0x4005_00B8) | bit Symbol  |     |    |    |
|                         | Read/Write  |     |    |    |
| VEIA01<br>(0x4005_0150) | After reset |     |    |    |
|                         | Function    |     |    |    |
|                         | 15          | ... | 1  | 0  |
|                         | bit Symbol  |     |    |    |
|                         | Read/Write  |     |    |    |
|                         | After reset |     |    |    |
|                         | Function    |     |    |    |

|                         |             |     |    |    |
|-------------------------|-------------|-----|----|----|
|                         | 31          | ... | 17 | 16 |
| VEIB00<br>(0x4005_00BC) | bit Symbol  |     |    |    |
|                         | Read/Write  |     |    |    |
| VEIB01<br>(0x4005_0154) | After reset |     |    |    |
|                         | Function    |     |    |    |
|                         | 15          | ... | 1  | 0  |
|                         | bit Symbol  |     |    |    |
|                         | Read/Write  |     |    |    |
|                         | After reset |     |    |    |
|                         | Function    |     |    |    |

|                         |             |     |    |    |
|-------------------------|-------------|-----|----|----|
|                         | 31          | ... | 17 | 16 |
| VEIC00<br>(0x4005_00C0) | bit Symbol  |     |    |    |
|                         | Read/Write  |     |    |    |
| VEIC01<br>(0x4005_0158) | After reset |     |    |    |
|                         | Function    |     |    |    |
|                         | 15          | ... | 1  | 0  |
|                         | bit Symbol  |     |    |    |
|                         | Read/Write  |     |    |    |
|                         | After reset |     |    |    |
|                         | Function    |     |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

- <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".

15.4.3.17 Current ADC Result Registers

|                           |             |     |    |    |
|---------------------------|-------------|-----|----|----|
|                           | 31          | ... | 17 | 16 |
| VEIAADC0<br>(0x4005_00C4) | bit Symbol  |     |    |    |
|                           | Read/Write  |     |    |    |
| VEIAADC1<br>(0x4005_015C) | After reset |     |    |    |
|                           | Function    |     |    |    |
|                           | 15          | ... | 1  | 0  |
|                           | bit Symbol  |     |    |    |
|                           | Read/Write  |     |    |    |
|                           | After reset |     |    |    |
|                           | Function    |     |    |    |

|                           |             |     |    |    |
|---------------------------|-------------|-----|----|----|
|                           | 31          | ... | 17 | 16 |
| VEIBADC0<br>(0x4005_00C8) | bit Symbol  |     |    |    |
|                           | Read/Write  |     |    |    |
| VEIBADC1<br>(0x4005_0160) | After reset |     |    |    |
|                           | Function    |     |    |    |
|                           | 15          | ... | 1  | 0  |
|                           | bit Symbol  |     |    |    |
|                           | Read/Write  |     |    |    |
|                           | After reset |     |    |    |
|                           | Function    |     |    |    |

|                           |             |     |    |    |
|---------------------------|-------------|-----|----|----|
|                           | 31          | ... | 17 | 16 |
| VEICADC0<br>(0x4005_00CC) | bit Symbol  |     |    |    |
|                           | Read/Write  |     |    |    |
| VEICADC1<br>(0x4005_0164) | After reset |     |    |    |
|                           | Function    |     |    |    |
|                           | 15          | ... | 1  | 0  |
|                           | bit Symbol  |     |    |    |
|                           | Read/Write  |     |    |    |
|                           | After reset |     |    |    |
|                           | Function    |     |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

- <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".

15.4.3.18 Supply Voltage Register

|                         |             |                               |    |    |
|-------------------------|-------------|-------------------------------|----|----|
|                         | 31          | ...                           | 17 | 16 |
| VEVDC0<br>(0x4005_00D0) | bit Symbol  | VDC                           |    |    |
|                         | Read/Write  | R                             |    |    |
| VEVDC1<br>(0x4005_0168) | After reset | 0x0000                        |    |    |
|                         | Function    | -                             |    |    |
|                         | 15          | ...                           | 1  | 0  |
|                         | bit Symbol  | VDC                           |    |    |
|                         | Read/Write  | R/W                           |    |    |
|                         | After reset | 0x0000                        |    |    |
|                         | Function    | Supply voltage: 0x0000-0xFFFF |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<VDC> Supply voltage (16-bit fixed-point data: 0 to 1.0)

The actual voltage value is:

$$\text{VDC value} \div \text{Max\_V value} \times 2^{15}$$

\* Max\_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC)  $\times 2^{12}$

15.4.3.19 dq Current Registers

|                        |             |                                       |   |   |
|------------------------|-------------|---------------------------------------|---|---|
|                        | 31          | ...                                   | 1 | 0 |
| VEID0<br>(0x4005_00D4) | bit Symbol  | ID                                    |   |   |
|                        | Read/Write  | R/W                                   |   |   |
| VEID1<br>(0x4005_016C) | After reset | 0x00000000                            |   |   |
|                        | Function    | d-axis current: 0x00000000-0xFFFFFFFF |   |   |

|                        |             |                                       |   |   |
|------------------------|-------------|---------------------------------------|---|---|
|                        | 31          | ...                                   | 1 | 0 |
| VEIQ0<br>(0x4005_00D8) | bit Symbol  | IQ                                    |   |   |
|                        | Read/Write  | R/W                                   |   |   |
| VEIQ1<br>(0x4005_0170) | After reset | 0x00000000                            |   |   |
|                        | Function    | q-axis current: 0x00000000-0xFFFFFFFF |   |   |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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:

$$\text{ID (or IQ) value} \div \text{Max\_I value} \times 2^{31}$$

\* Max\_I: (Phase current value [A] which corresponds to 1 LSB of ADC)  $\times 2^{11}$

15.4.3.20 PWM Duty Register

|                          |             |     |    |    |
|--------------------------|-------------|-----|----|----|
|                          | 31          | ... | 17 | 16 |
| VECMPU0<br>(0x4005_017C) | bit Symbol  |     |    |    |
|                          | Read/Write  |     |    |    |
|                          | After reset |     |    |    |
| VECMPU1<br>(0x4005_019C) | Function    |     |    |    |
|                          | 15          | ... | 1  | 0  |
|                          | bit Symbol  |     |    |    |
|                          | Read/Write  |     |    |    |
|                          | After reset |     |    |    |
|                          | Function    |     |    |    |

|                          |             |     |    |    |
|--------------------------|-------------|-----|----|----|
|                          | 31          | ... | 17 | 16 |
| VECMPV0<br>(0x4005_0180) | bit Symbol  |     |    |    |
|                          | Read/Write  |     |    |    |
| VECMPV1<br>(0x4005_01A0) | After reset |     |    |    |
|                          | Function    |     |    |    |
|                          | 15          | ... | 1  | 0  |
|                          | bit Symbol  |     |    |    |
|                          | Read/Write  |     |    |    |
|                          | After reset |     |    |    |
|                          | Function    |     |    |    |

|                          |             |     |    |    |
|--------------------------|-------------|-----|----|----|
|                          | 31          | ... | 17 | 16 |
| VECMPW0<br>(0x4005_0184) | bit Symbol  |     |    |    |
|                          | Read/Write  |     |    |    |
| VECMPW1<br>(0x4005_01A4) | After reset |     |    |    |
|                          | Function    |     |    |    |
|                          | 15          | ... | 1  | 0  |
|                          | bit Symbol  |     |    |    |
|                          | Read/Write  |     |    |    |
|                          | After reset |     |    |    |
|                          | Function    |     |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

- <VCMPU> PMD setting: PWM setting of U-phase
- <VCMPV> PMD setting: PWM setting of V-phase
- <VCMPW> PMD setting: PWM setting of W-phase

15.4.3.21 6-Phase Output Control Register

|                           |   |   |   |    |   |    |   |   |
|---------------------------|---|---|---|----|---|----|---|---|
|                           | 31  | ...   |   |    |   |    | 17  | 16  |
| VEOUTCR0<br>(0x4005_0188) | bit Symbol  |   |   |    |   |    |   |   |
|                           | Read/Write  |   |   |    |   |    |   |   |
|                           | After reset   |   |   |    |   |    |   |   |
|                           | Function  |   |   |    |   |    |   |   |
| VEOUTCR1<br>(0x4005_01A8) | 15  | 14  | 13  | 12 | 11  | 10 | 9   | 8   |
|                           | bit Symbol  |   |   |    |   |    |   | WPWM  |
|                           | Read/Write  |   |   |    |   |    |   | R/W   |
|                           | After reset   |   |   |    |   |    |   | 0   |
|                           | Function  |   |   |    |   |    |   | PWM of W-phase<br>0: ON/OFF output<br>1: PWM output |
|                           | 7   | 6   | 5   | 4  | 3   | 2  | 1   | 0   |
|                           | VPWM  | UPWM  | WOC   |    | VOC   |    | UOC   |   |
|                           | R/W   | R/W   | R/W   |    | R/W   |    | R/W   |   |
|                           | 0   | 0   | 00  |    | 00  |    | 00  |   |
|                           | PWM of V-phase<br>0: ON/OFF output<br>1: PWM output | PWM of U-phase<br>0: ON/OFF output<br>1: PWM output | Output control of W-phase<br>00: W0 OFF, Z0 OFF <sup>(*)</sup><br>01: W0 ON, Z0 OFF<br>10: W0 OFF, Z0 ON<br>11: W0 ON, Z0 ON<br>*W0 and Z0 are both ON when WPWM=1. |    | Output control of V-phase<br>00: V0 OFF, Y0 OFF <sup>(*)</sup><br>01: V0 ON, Y0 OFF<br>10: V0 OFF, Y0 ON<br>11: V0 ON, Y0 ON<br>*V0 and Y0 are both ON when VPWM=1. |    | Output control of U-phase<br>00: U0 OFF, X0 OFF <sup>(*)</sup><br>01: U0 ON, X0 OFF<br>10: U0 OFF, X0 ON<br>11: U0 ON, X0 ON<br>*U0 and X0 are both ON when UPWM=1. |   |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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<br>inverted output | PWMU<br>output          |
| 1       | 11  | PWMU<br>output          | PWMU<br>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<br>inverted output | PWMV<br>output          |
| 1       | 11  | PWMV<br>output          | PWMV<br>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<br>inverted output | PWMW<br>output          |
| 1       | 11  | PWMW<br>output          | PWMW<br>inverted output |

\* The table shows only those combinations that are used in the VE.



15.4.3.22 VETRGCRC Register

|                            |             |                    |    |    |
|----------------------------|-------------|--------------------|----|----|
|                            | 31          | ...                | 17 | 16 |
| VETRGCRC0<br>(0x4005_0098) | bit Symbol  |                    |    |    |
|                            | Read/Write  | R                  |    |    |
| VETRGCRC1<br>(0x4005_0130) | After reset | 0x0000             |    |    |
|                            | Function    | -                  |    |    |
|                            | 15          | ...                | 1  | 0  |
|                            | bit Symbol  | TRGCRC             |    |    |
|                            | Read/Write  | R/W                |    |    |
|                            | After reset | 0x0000             |    |    |
|                            | Function    | Trigger correction |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<TRGCRC>: Used to correct the synchronizing trigger timing. The value to be set is:  
 Correction time [s] ÷ PWM counter clock frequency [s]

15.4.3.23 VETRGCMP Register

|                             |             |   |    |    |
|-----------------------------|-------------|---|----|----|
|                             | 31          | ...   | 17 | 16 |
| VETRGCMP00<br>(0x4005_018C) | bit Symbol  |   |    |    |
|                             | Read/Write  | R   |    |    |
| VETRGCMP01<br>(0x4005_01AC) | After reset | 0x0000  |    |    |
|                             | Function    | -   |    |    |
|                             | 15          | ...   | 1  | 0  |
|                             | bit Symbol  | VTRGCMPO  |    |    |
|                             | Read/Write  | R/W   |    |    |
|                             | After reset | 0x0000  |    |    |
|                             | Function    | Trigger timing setting<br>0x0000: Prohibited<br>0x0001 to (MDPRD value -1): Trigger timing<br>MDPRD value to 0xFFFF: Prohibited |    |    |
|                             | 31          | ...   | 17 | 16 |
| VETRGCMP10<br>(0x4005_0190) | bit Symbol  |   |    |    |
|                             | Read/Write  | R   |    |    |
| VETRGCMP11<br>(0x4005_01B0) | After reset | 0x0000  |    |    |
|                             | Function    | -   |    |    |
|                             | 15          | ...   | 1  | 0  |
|                             | bit Symbol  | VTRGCMPI  |    |    |
|                             | Read/Write  | R/W   |    |    |
|                             | After reset | 0x0000  |    |    |
|                             | Function    | Trigger timing setting<br>0x0000: Prohibited<br>0x0001 to (MDPRD value -1): Trigger timing<br>MDPRD value to 0xFFFF: Prohibited |    |    |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<VTRGCMPO> PMD setting: Specifies the trigger timing for sampling ADC in synchronization with PMD.

<VTRGCMPI> 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).

15.4.3.24 VETRGSEL Register

|                            |             |          |   |   |   |   |  |   |   |  |
|----------------------------|-------------|----------|---|---|---|---|--|---|---|--|
|                            | 31          | ...      |   |   |   | 9 | 8  |   |   |  |
| VETRGSEL0<br>(0x4005_0194) | bit Symbol  |          |   |   |   |   |  |   |   |  |
|                            | Read/Write  | R        |   |   |   |   |  |   |   |  |
| VETRGSEL1<br>(0x4005_01B4) | 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<br>0-5: Output trigger number<br>6-7: Prohibited |   |   |  |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<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).

15.4.3.25 VEEMGRS Register

|                           |             |          |     |  |  |   |   |   |  |
|---------------------------|-------------|----------|-----|--|--|---|---|---|--|
|                           | 31          | ...      |     |  |  | 9 | 8   |   |  |
| VEEMGRS0<br>(0x4005_0198) | bit Symbol  |          |     |  |  |   |   |   |  |
|                           | Read/Write  | R        |     |  |  |   |   |   |  |
| VEEMGRS1<br>(0x4005_01B8) | After reset | 0x000000 |     |  |  |   |   |   |  |
|                           | Function    | -        |     |  |  |   |   |   |  |
|                           |             | 7        | ... |  |  | 2 | 1   | 0 |  |
|                           | bit Symbol  |          |     |  |  |   | EMGRS   |   |  |
|                           | Read/Write  | R        |     |  |  |   | R/W   |   |  |
|                           | After reset | 0000000  |     |  |  |   | 0   |   |  |
|                           | Function    | -        |     |  |  |   | EMG return<br>0: Nop<br>1: EMG return command |   |  |

The upper and lower register names correspond to channel 0 and channel 1 respectively.

<EMGRS> PMD setting : EMG return command for returning from the EMG state

## 15.5 Description of Operations

### 15.5.1 Schedule Management

Figure 15-4 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.

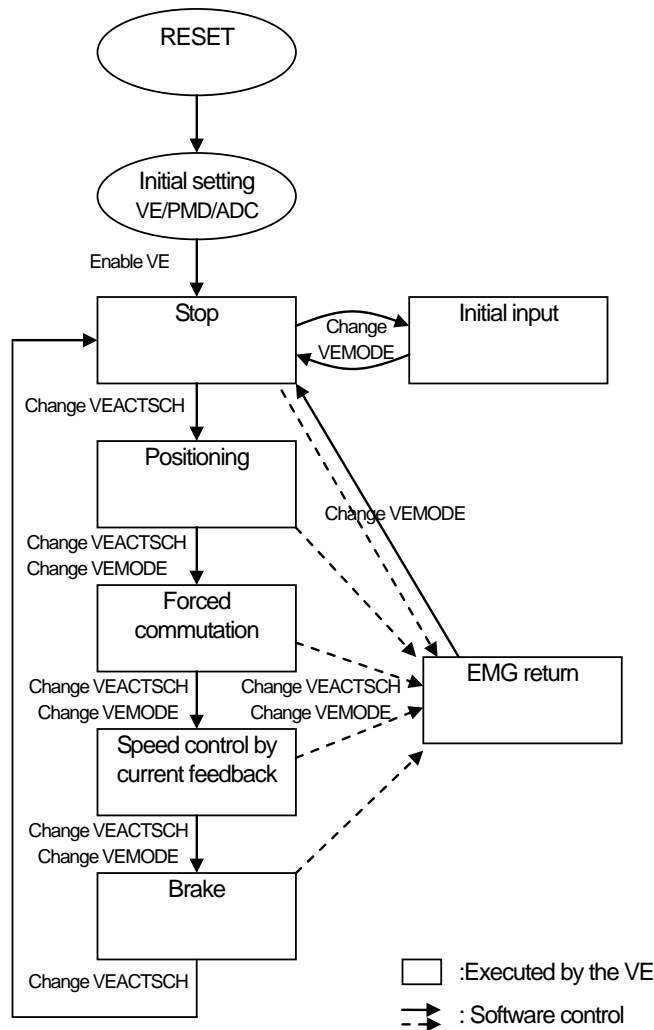


Figure 15-4 Example of Motor Control Flow

|                                   |   |
|-----------------------------------|---|
| 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.  |

### 15.5.1.1 Schedule Control

The VECTSCH 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 15-5 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 15-6).

Table 15-5 Tasks To Be Executed in Each Schedule

| Schedule Selection<br>(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 15-6 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.

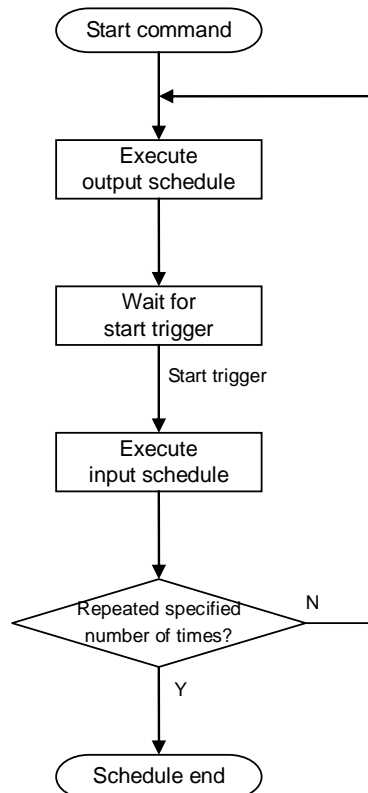


Figure 15-5 Schedule Execution Flow

### 15.5.1.2 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.

### 15.5.2 Summary of Tasks

Table 15-7 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 15-7 List of Tasks

|                    | Task                              | Description  | Task Number |
|--------------------|-----------------------------------|--|-------------|
| Output<br>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 $\alpha\beta$ 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<br>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 $\alpha\beta$ coordinates to dq coordinates.                    | 4           |

### 15.5.2.1 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    | VEIDn         | d-axis current                    | 32-bit fixed-point data (31 fractional bits) |
|          | VEIDREFn      | Reference value of d-axis current | 16-bit fixed-point data (15 fractional bits) |
|          | VECIDKPn      | Proportional coefficient          | 16-bit data                                  |
|          | VECIDKIn      | Integral coefficient              | 16-bit data                                  |
| Output   | VEVDn         | d-axis voltage                    | 32-bit fixed-point data (31 fractional bits) |
| Internal | VDIn          | Integral term of d-axis voltage   | 64-bit fixed-point data (63 fractional bits) |

n = channel number

#### 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    | VEIQn         | q-axis current                    | 32-bit fixed-point data (31 fractional bits) |
|          | VEIQREFn      | Reference value of q-axis current | 16-bit fixed-point data (15 fractional bits) |
|          | VECIQKPn      | Proportional coefficient          | 16-bit data                                  |
|          | VECIQKIn      | Integral coefficient              | 16-bit data                                  |
| Output   | VEVQn         | q-axis voltage                    | 32-bit fixed-point data (31 fractional bits) |
| Internal | VQIn          | Integral term of q-axis voltage   | 64-bit fixed-point data (63 fractional bits) |

n = channel number

15.5.2.2 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]

$$VETHEATA = (VEOMEGA \times VETPWM + VETHEATA \times 2^{31}) / 2^{31}$$

: Integration of rotation speed.  
Only when phase interpolation is enabled.

|        | Register Name | Description                |   |
|--------|---------------|----------------------------|---|
| Input  | VETHETAn      | Phase $\theta$             | 16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)  |
|        | VEOMEGAn      | Rotation speed             | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
|        | VETPWM        | PWM period rate            | 16-bit data   |
|        | VEMODEn       | Phase interpolation enable | Mode settings   |
| Output | VETHETAn      | Phase $\theta$             | 16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)  |

n = channel number

ii) SIN/COS computation

[Equations]

$$VESINM = VESIN$$

: Saves previous value (for input processing).

$$VECOSM = VECOS$$

: Saves previous value (for input processing).

$$VESIN = \sin(VETHETA \times \pi / 2^{15}) \times 2^{15}$$

: SIN/COS computation

$$VECOS = \sin((VETHETA + 0x4000) \times \pi / 2^{15}) \times 2^{15}$$

|        | Register Name | Description              |   |
|--------|---------------|--------------------------|---|
| Input  | VETHETAn      | Phase $\theta$           | 16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)  |
| Output | VESINn        | Sine value at $\theta$   | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
|        | VECOSn        | Cosine value at $\theta$ | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
|        | VESINMn       | Previous sine value      | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
|        | VECOSMn       | Previous cosine value    | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |

n = channel number



15.5.2.3 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  $V_\alpha$  and  $V_\beta$  from  $V_d$ ,  $V_q$  in SIN and COS.

The 2-phase-to-3-phase conversion performs segmentation by using  $V_\alpha$  and  $V_\beta$  and performs space vector conversion to calculate  $V_a$ ,  $V_b$  and  $V_c$ .

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]

$VETMPREG3 = (VECOS \times VEVD - VESIN \times VEVQ) / 2^{15}$  : Calculates  $V_\alpha$ .

$VETMPREG4 = (VESIN \times VEVD + VECOS \times VEVQ) / 2^{15}$  : Calculates  $V_\beta$ .

|        | Register Name | Description              |   |
|--------|---------------|--------------------------|---|
| Input  | VEVDn         | d-axis voltage           | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VEVQn         | q-axis voltage           | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VESINn        | Sine value at $\theta$   | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
|        | VECOSn        | Cosine value at $\theta$ | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
| Output | VETMPREG3     | $\alpha$ -axis voltage   | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VETMPREG4     | $\beta$ -axis voltage    | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |

n = channel number

(2) 2-phase-to-3-phase conversion (space vector conversion)

i) Segmentation

[Equations]

$VESECTORM = VESECTOR$  : Saves previous sector.

```

if( $V_\alpha \geq 0$  &  $V_\beta \geq 0$ )
  if( $|V_\alpha| \geq |V_\beta| \div \sqrt{3}$ )
    if( $|V_\alpha| \div \sqrt{3} \geq |V_\beta|$ ) SECTOR=0
    else SECTOR=1
  else SECTOR=2
else if( $V_\alpha < 0$  &  $V_\beta \geq 0$ )
  if( $|V_\alpha| < |V_\beta| \div \sqrt{3}$ ) SECTOR=3
  else if( $|V_\alpha| \div \sqrt{3} < |V_\beta|$ ) SECTOR=4
  else SECTOR=5
else if( $V_\alpha < 0$  &  $V_\beta < 0$ )
  if( $|V_\alpha| \geq |V_\beta| \div \sqrt{3}$ )
    if( $|V_\alpha| \div \sqrt{3} \geq |V_\beta|$ ) SECTOR=6
    else SECTOR=7
  else SECTOR=8
else if( $V_\alpha \geq 0$  &  $V_\beta < 0$ )
  if( $|V_\alpha| < |V_\beta| \div \sqrt{3}$ ) SECTOR=9
  else if( $|V_\alpha| \div \sqrt{3} < |V_\beta|$ ) SECTOR=10
  else SECTOR=11
    
```

|        | Register Name | Description            |   |
|--------|---------------|------------------------|---|
| Input  | VETMPREG3     | $\alpha$ -axis voltage | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VETMPREG4     | $\beta$ -axis voltage  | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
| Output | VESECTORn     | Sector                 | 4-bit data  |
|        | VESECTORMn    | Previous sector        | 4-bit data  |

n = channel number

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     | $\alpha$ -axis voltage | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VETMPREG4     | $\beta$ -axis voltage  | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VEVDCn        | Supply voltage         | 16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)  |
|        | VESECTORn     | Sector                 | 4-bit data  |
|        | VEFMODEn      | 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)  |

n = channel number

### 15.5.2.4 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)  |
|        | VEMDPRDn      | PWM period   | 16-bit data (PMD PWM period )                             |
|        | VESECTORn     | Sector   | 4-bit data  |
|        | VEOMEGAn      | Rotation speed   | 16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits) |
|        | VEFPWMCHGn    | PWM shift switching reference                              | 16-bit fixed-point data (0 to 1.0, 15 fractional bits)    |
|        | VEMODEn       | Output control operation                                   | Mode settings   |
|        | VEFMODEn      | PMD channel/ shift enable/ modulation mode/ detection mode | Mode settings   |
| Output | VECMPUn       | PMD U-phase PMW setting                                    | 16- bit data (0 to MDPRD value)                           |
|        | VECMPVn       | PMD V-phase PWM setting                                    | 16-bit data (0 to MDPRD value)                            |
|        | VECMPWn       | PMD W-phase PWM setting                                    | 16-bit data (0 to MDPRD value)                            |
|        | VEOUTCRn      | PMD output control setting                                 | 9-bit setting   |
|        | VEEMGRSn      | PMD EMG return   | 1-bit setting   |
|        | VEMCTLFn      | Shift switching flag                                       | Status  |

n = channel number

### 15.5.2.5 Trigger Generation

The trigger generation unit calculates the trigger timing from the PWM setting values (VECMPI, VECMPV and VECMPW) as appropriate to the current detection method, and sets the VETRGCMP0 and VETRGCMP1 registers.

|        | Register Name         | Description  |                                |
|--------|-----------------------|--|--------------------------------|
| Input  | VECMPI <sub>n</sub>   | PMD U-phase PWM setting                                    | 16-bit data (0 to MDPRD value) |
|        | VECMPV <sub>n</sub>   | PMD V-phase PWM setting                                    | 16-bit data (0 to MDPRD value) |
|        | VECMPW <sub>n</sub>   | PMD W-phase PWM setting                                    | 16-bit data (0 to MDPRD value) |
|        | VEMDPRD <sub>n</sub>  | PWM period setting   | 16-bit data (PMD PWM period)   |
|        | VETADC                | AD conversion time   | 16-bit data (0 to MDPRD value) |
|        | VETRGCRCh             | Trigger correction value                                   | 16-bit data (0 to MDPRD value) |
|        | VESECTOR <sub>n</sub> | Sector   | 4-bit data                     |
|        | VEMODE <sub>n</sub>   | Output control operation                                   | Mode settings                  |
|        | VEFMODE <sub>n</sub>  | PMD channel/ shift enable/ modulation mode/ detection mode | Mode settings                  |
|        | VEMCTLF <sub>n</sub>  | 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) |
|        | VETRGSEL <sub>n</sub> | PMD trigger selection                                      | 3-bit data                     |

n = channel number

### 15.5.2.6 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  |
|          | VESECTORMn    | Sector information  | 4-bit data  |
|          | VEMODEn       | Zero-current detection  | Mode settings   |
|          | VEFMODEn      | PMD channel /current detection mode /<br>ADC unit /shift enable | Mode settings   |
|          | VEMCTLFn      | Shift switching flag  | Status  |
| Output   | VEVDCn        | 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 | IA0n          | a-phase zero-current conversion result                          | 16-bit data (The upper 12 bits are used.)                 |
|          | IB0n          | b-phase zero-current conversion result                          | 16-bit data (The upper 12 bits are used.)                 |
|          | IC0n          | c-phase zero-current conversion result                          | 16-bit data (The upper 12 bits are used.)                 |
|          | IAADCn        | a-phase current conversion result                               | 16-bit data (The upper 12 bits are used.)                 |
|          | IBADCn        | b-phase current conversion result                               | 16-bit data (The upper 12 bits are used.)                 |
|          | ICADCn        | c-phase current conversion result                               | 16-bit data (The upper 12 bits are used.)                 |

n = channel number

### 15.5.2.7 Input Current Conversion (Phase Conversion/Coordinate axis Conversion)

Input current conversion is comprised of 3-phase-to-2-phase conversion and  $\alpha\beta$ -to-dq coordinate axis conversion.

The 3-phase-to-2-phase conversion calculates  $I_\alpha$  and  $I_\beta$  from  $I_a$ ,  $I_b$  and  $I_c$ .

The  $\alpha\beta$ -to-dq coordinate axis conversion calculates  $I_d$  and  $I_q$  from  $I_\alpha$ ,  $I_\beta$ , VESINM and VECOSM.

#### (1) 3-phase-to-2-phase conversion

[Equations]

$$\text{VETMPREG3} = \text{VETMPREG0} \quad : \text{Calculates } I_\alpha.$$

$$\text{VETMPREG4} = 1/\sqrt{3} \times \text{VETMPREG1} - 1/\sqrt{3} \times \text{VETMPREG2} \quad : \text{Calculates } I_\beta.$$

|        | 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     | $\alpha$ -axis current | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VETMPREG4     | $\beta$ -axis current  | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |

n = channel number

#### (2) $\alpha\beta$ -to-dq coordinate conversion

[Equations]

$$\text{VEID} = \text{VECOSM} \times \text{VETMPREG3} + \text{VESINM} \times \text{VETMPREG4} \quad : \text{Calculates } I_d.$$

$$\text{VEIQ} = -\text{VESINM} \times \text{VETMPREG3} + \text{VECOSM} \times \text{VETMPREG4} \quad : \text{Calculates } I_q.$$

|        | Register Name | Description              |   |
|--------|---------------|--------------------------|---|
| Input  | VETMPREG3     | $\alpha$ -axis current   | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VETMPREG4     | $\beta$ -axis current    | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VESINMn       | Sine value at $\theta$   | 16-bit fixed-point data (-1.0 ~ 1.0, 15 fractional bits)  |
|        | VECOSMn       | Cosine value at $\theta$ | 16-bit fixed-point data (-1.0 ~ 1.0, 15 fractional bits)  |
| Output | VEIDn         | d-axis current           | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |
|        | VEIQn         | q-axis current           | 32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits) |

n = channel number

## 15.6 Combinations of VE Channel, ADC Unit and PMD Channel

Table 15-8 and Table 15-9 show the combinations of AD unit and PMD channel that can be used with each Vector Engine channel depending on the current detection mode to be used.

The Vector Engine calculates the stored value in AD conversion result register 0 to 2 (ADxREG0 to 2) as a current data and calculates the stored value in AD conversion result register 3 (ADxREG3) as a voltage data. Therefore, please specify it with proper setting, referring to Table 15-9.

(x=A or B)

Table 15-8 Combination of VE and PMD

| Vector Engine | PMD       |
|---------------|-----------|
| Channel 0     | Channel 0 |
| Channel 1     | Channel 1 |

Table 15-9 Combination of VE and ADC

| Vector Engine |                               |                           | ADC Unit A     |                |         |          | ADC Unit B     |                |         |          |
|---------------|-------------------------------|---------------------------|----------------|----------------|---------|----------|----------------|----------------|---------|----------|
| Channel       | VEMODE(Note 1)                |                           | ADAREG0        | ADAREG1        | ADAREG2 | ADAREG3  | ADBREG0        | ADBREG1        | ADBREG2 | ADBREG3  |
|               | Current detection<br><IDMODE> | ADC selection<br><ADCSEL> |                |                |         |          |                |                |         |          |
| 0             | 0x                            | 00                        | Current data 1 | Current data 2 | Note 2  | VDC data | -              | -              | -       | -        |
|               |                               | 1x                        | Current data 1 | -              | Note 2  | VDC data | Current data 2 | -              | -       | -        |
|               | 1x                            | 00                        | Current data 1 | Current data 2 | -       | VDC data | -              | -              | -       | -        |
| 1             | 0x                            | 01                        | -              | -              | -       | -        | Current data 1 | Current data 2 | Note 2  | VDC data |
|               |                               | 1x                        | -              | Current data 2 | -       | -        | -              | Current data 1 | Note 2  | VDC data |
|               | 1x                            | 01                        | -              | -              | -       | -        | Current data 1 | Current data 2 | -       | VDC data |

Note 1: Please do not use the combination of VE and ADC which is not allowed in the table.

Note 2: Specifying the phase information to the register is necessary. However the AD conversion result of its register is not used for calculation.

### 16. Motor Control Circuit (PMD: Programmable Motor Driver)

The TMPM370 contains a two-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.

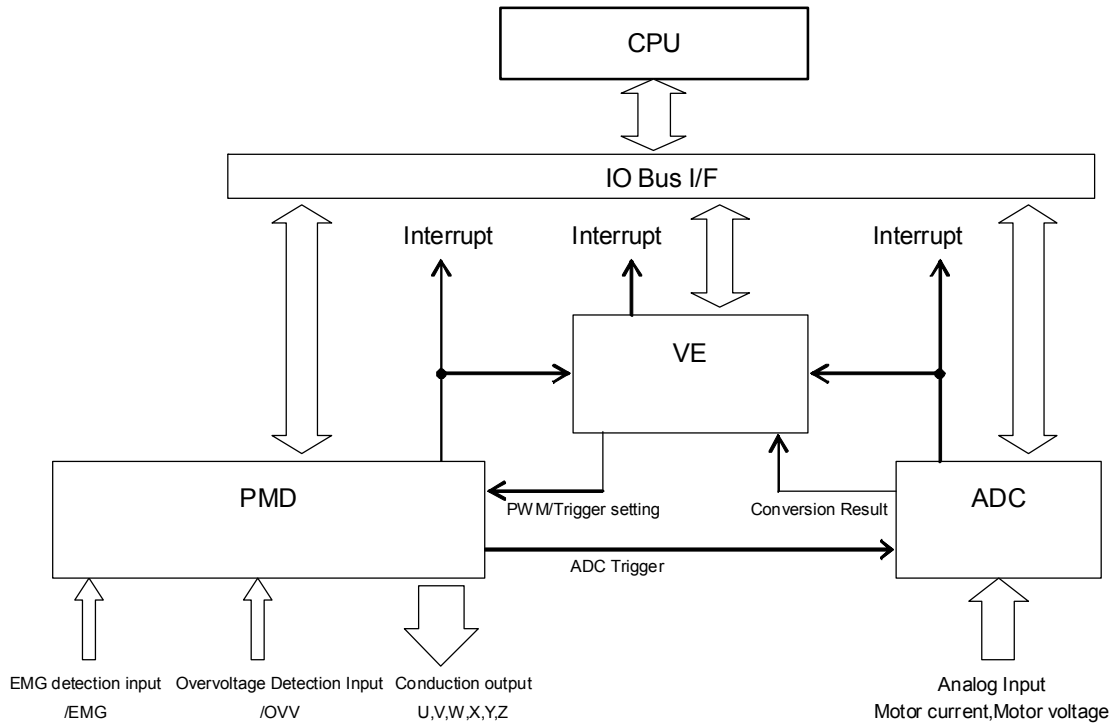


Figure 16-1 Motor Control-related Block Constitution

## 16.1 PMD Input/Output Signals

The table below shows the signals that are input to and output from each PMD.

Table 16-1 Input/Output Signals

| Channel | Pin Name                      | PMD Signal Name | Description      |
|---------|-------------------------------|-----------------|------------------|
| PMD0    | PC7/ $\overline{\text{OVV0}}$ | OVV 0           | OVV state signal |
|         | PC6/ $\overline{\text{EMG0}}$ | EMG 0           | EMG state signal |
|         | PC0/UO0                       | UO 0            | U-phase output   |
|         | PC1/XO0                       | X O0            | X-phase output   |
|         | PC2/VO0                       | V O0            | V-phase output   |
|         | PC3/YO0                       | Y O0            | Y- phase output  |
|         | PC4/WO0                       | W O0            | W-phase output   |
|         | PC5/ZO0                       | Z O0            | Z-phase output   |
| PMD1    | PG7/ $\overline{\text{OVV1}}$ | OVV 1           | OVV state signal |
|         | PG6/ $\overline{\text{EMG1}}$ | 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   |



## 16.2 PMD Registers

The table below shows the registers related to the PMD. (The Address column shows a PMD0 address first followed by a PMD1 address on the next line.)

Table 16-2 PMD Registers

| Address                    | Mnemonic                   | Register Name                        |
|----------------------------|----------------------------|--------------------------------------|
| 0x4005 0400<br>0x4005 0480 | PMD0MDEN<br>PMD1MDEN       | PMD Enable Register                  |
| 0x4005 0404<br>0x4005 0484 | PMD0PORTMD<br>PMD1PORTMD   | Port Output Mode Register            |
| 0x4005 0408<br>0x4005 0488 | PMD0MDCR<br>PMD1MDCR       | PMD Control Register                 |
| 0x4005 040C<br>0x4005 048C | PMD0CNTSTA<br>PMD1CNTSTA   | PWM Counter Status Register          |
| 0x4005 0410<br>0x4005 0490 | PMD0MDCNT<br>PMD1MDCNT     | PWM Counter Register                 |
| 0x4005 0414<br>0x4005 0494 | PMD0MDPRD<br>PMD1MDPRD     | PWM Period Register                  |
| 0x4005 0418<br>0x4005 0498 | PMD0CMPU<br>PMD1CMPU       | PMD Compare U Register               |
| 0x4005 041C<br>0x4005 049C | PMD0CMPV<br>PMD1CMPV       | PMD Compare V Register               |
| 0x4005 0420<br>0x4005 04A0 | PMD0CMPW<br>PMD1CMPW       | PMD Compare W Register               |
| 0x4005 0424<br>0x4005 04A4 | PMD0MODESEL<br>PMD1MODESEL | Mode Select Register                 |
| 0x4005 0428<br>0x4005 04A8 | PMD0MDOUT<br>PMD1MDOUT     | PMD Output Control Register          |
| 0x4005 042C<br>0x4005 04AC | PMD0MDPOT<br>PMD1MDPOT     | PMD Output Setting Register          |
| 0x4005 0430<br>0x4005 04B0 | PMD0EMGREL<br>PMD1EMGREL   | EMG Release Register                 |
| 0x4005 0434<br>0x4005 04B4 | PMD0EMGCR<br>PMD1EMGCR     | EMG Control Register                 |
| 0x4005 0438<br>0x4005 04B8 | PMD0EMGSTA<br>PMD1EMGSTA   | EMG Status Register                  |
| 0x4005 043C<br>0x4005 04BC | PMD0OVVCR<br>PMD1OVVCR     | OVV Control Register                 |
| 0x4005 0440<br>0x4005 04C0 | PMD0OVVSTA<br>PMD1OVVSTA   | OVV Status Register                  |
| 0x4005 0444<br>0x4005 04C4 | PMD0DTR<br>PMD1DTR         | Dead Time Register                   |
| 0x4005 0448<br>0x4005 04C8 | PMD0TRGCMP0<br>PMD1TRGCMP0 | Trigger Compare 0 Register           |
| 0x4005 044C<br>0x4005 04CC | PMD0TRGCMP1<br>PMD1TRGCMP1 | Trigger Compare 1 Register           |
| 0x4005 0450<br>0x4005 04D0 | PMD0TRGCMP2<br>PMD1TRGCMP2 | Trigger Compare 2 Register           |
| 0x4005 0454<br>0x4005 04D4 | PMD0TRGCMP3<br>PMD1TRGCMP3 | Trigger Compare 3 Register           |
| 0x4005 0458<br>0x4005 04D8 | PMD0TRGCR<br>PMD1TRGCR     | Trigger Control Register             |
| 0x4005 045C<br>0x4005 04DC | PMD0TRGMD<br>PMD1TRGMD     | Trigger Output Mode Setting Register |
| 0x4005 0460<br>0x4005 04E0 | PMD0TRGSEL<br>PMD1TRGSEL   | Trigger Output Select Register       |
| 0x4005 047C<br>0x4005 04FC | Reserved                   | -                                    |

Note: Do not access to "Reserved" address.

16.3 PMD Circuit

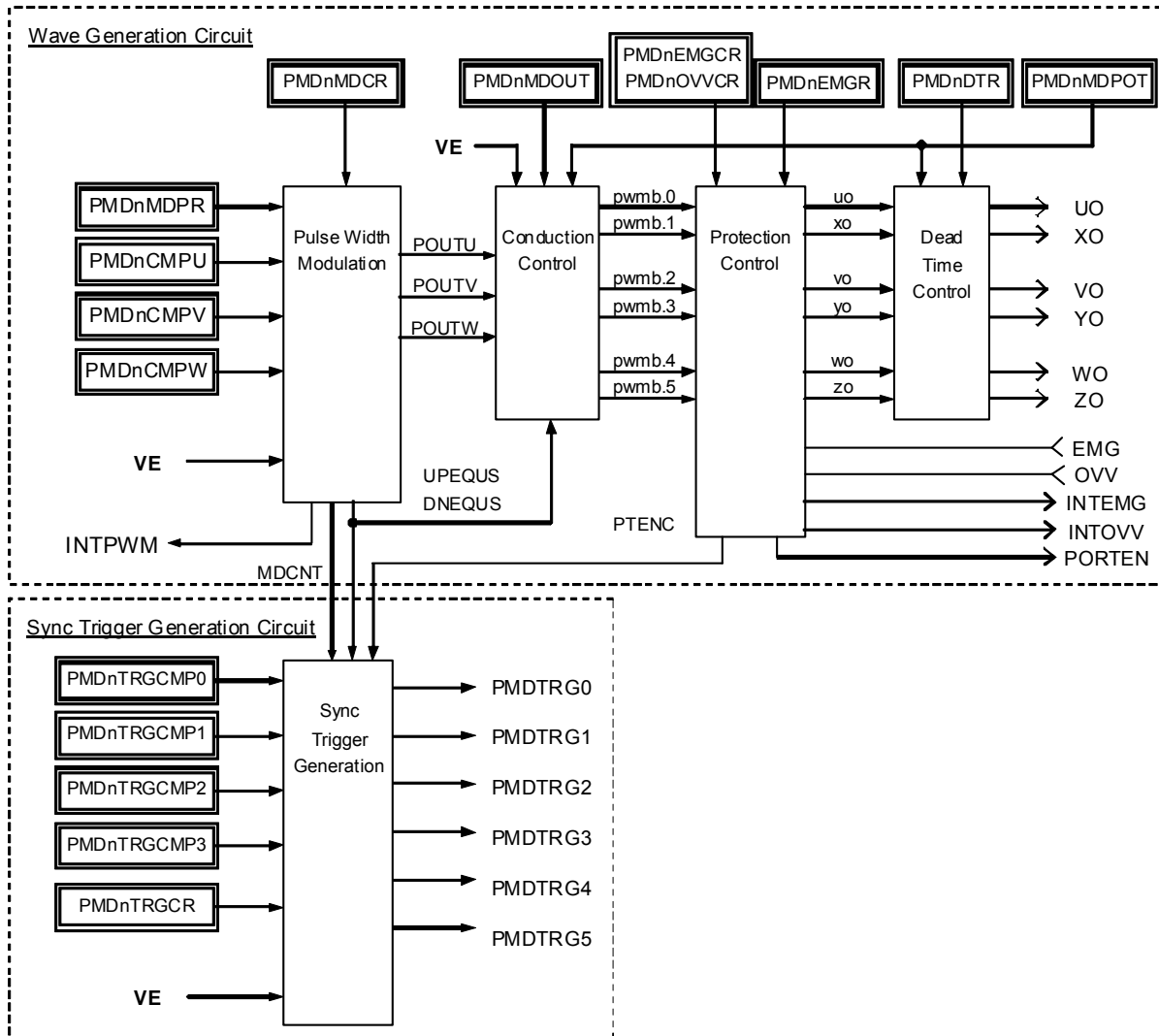


Figure 16-2 Block diagram of PMD Circuit

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.

### 16.3.1 PMD mode control registers

#### 16.3.1.1 PMD Enable Register (PMDnMDEN)

(PMD0:0x4005 0400, PMD1:0x4005 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.

#### 16.3.1.2 Port Output Mode Register (PMDnPORTMD)

(PMD0:0x4005 0404, PMD1:0x4005 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.

### 16.3.1.3 Mode Select Register (PMDnMODESEL) (PMD0:0x4005 0424, PMD1:0x4005 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  | -   | -   | -   | -   | -   | -   | -   | MDESEL |
| 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      |

<MDESEL>: 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.

16.3.2 Pulse Width Modulation Circuit

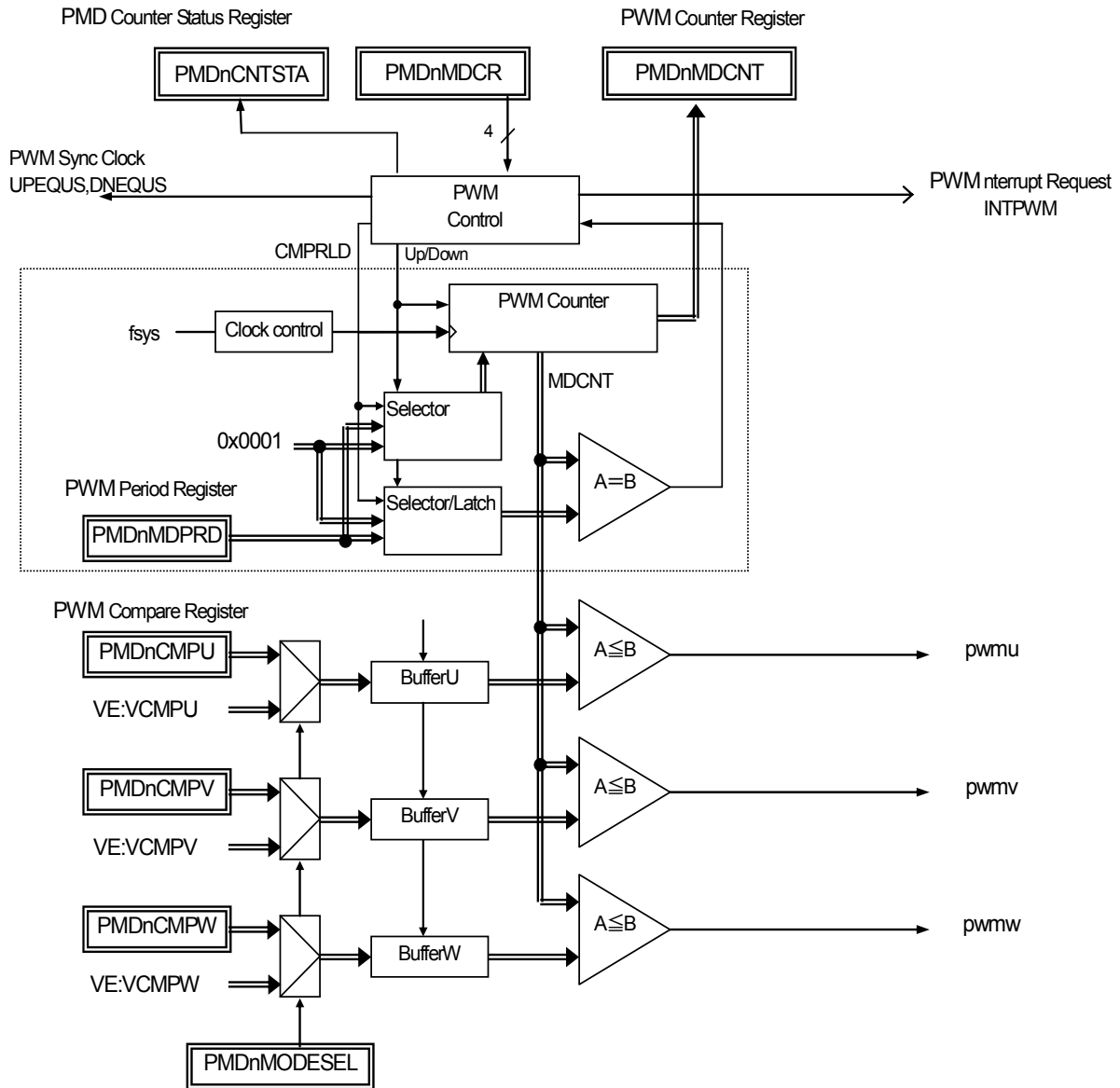


Figure 16-3 Pulse Width Modulation Circuit

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 (PMDnCMPU/V/W) and the carrier wave generated by the PWM counter (PMDnMDCNT) 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.

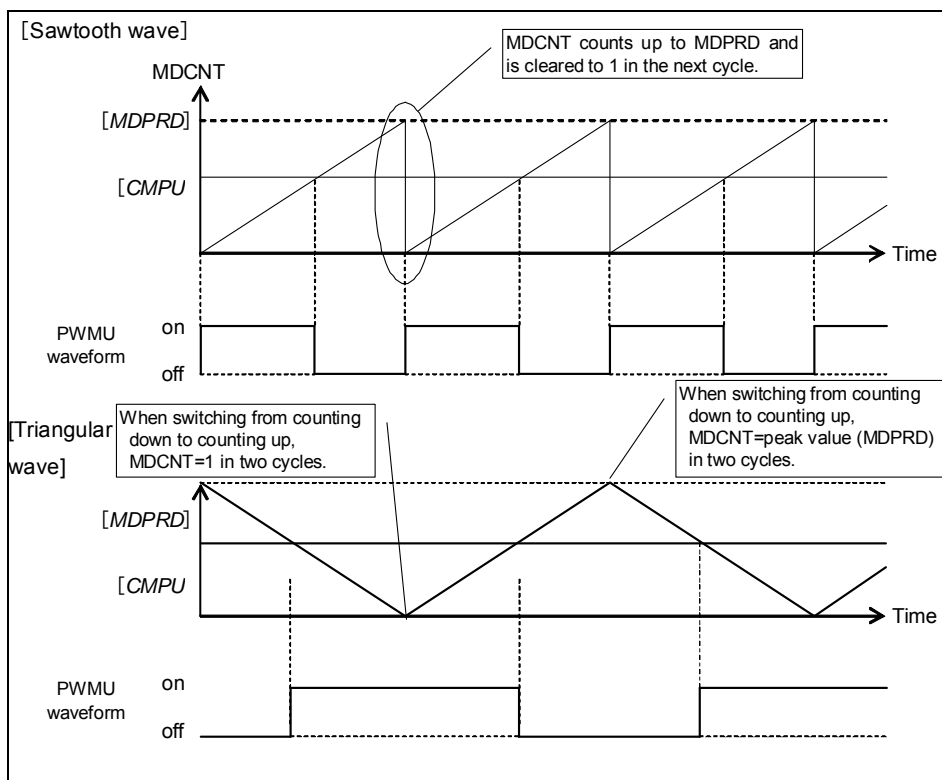


Figure 16-4 PWM Waveforms

(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.

### 16.3.2.1 PMD Control Register (PMDnMDCR) (PMD0:0x4005 0408, PMD1:0x4005 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 16-4.)

<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  $f_{sys}=80$  MHz.

\* Sawtooth wave: 12.5 ns, triangular wave: 25 ns

When <PWMCK>=1, the PWM counter operates with a resolution of 50 ns at  $f_{sys}=80$  MHz.

\* Sawtooth wave: 50 ns, triangular wave: 100 ns



### 16.3.2.2 PWM Counter Status Register (PMDnCNTSTA) (PMD0:0x4005 040C, PMD1:0x4005 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.

### 16.3.2.3 PWM Counter Register (PMDnMDCNT) (PMD0:0x4005 0410, PMD1:0x4005 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  $f_{sys} = 80$  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  $MDPRD$

### 16.3.2.4 PWM Period Register (PMDnMDPRD) (PMD0:0x4005 0414, PMD1:0x4005 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  $\geq$  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.

16.3.2.5 PWM Compare Registers (PMDnCMPU, PMDnCMPV, PMDnCMPW)  
 PMD0(0x4005 0418-041B),PMD1( 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  | CMPU0,1 |     |     |     |     |     |     |     |
| Read/Write  | R/W     |     |     |     |     |     |     |     |
| After reset | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|             | 7       | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit Symbol  | CMPU0,1 |     |     |     |     |     |     |     |
| Read/Write  | R/W     |     |     |     |     |     |     |     |
| After reset | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

PMD0(0x4005 041C-041F),PMD1( 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  | CMPV0,1 |     |     |     |     |     |     |     |
| Read/Write  | R/W     |     |     |     |     |     |     |     |
| After reset | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|             | 7       | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit Symbol  | CMPV0,1 |     |     |     |     |     |     |     |
| Read/Write  | R/W     |     |     |     |     |     |     |     |
| After reset | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

PMD0(0x4005 0420-0423),PMD1( 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  | CMPW0,1 |     |     |     |     |     |     |     |
| Read/Write  | R/W     |     |     |     |     |     |     |     |
| After reset | 0       | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|             | 7       | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit Symbol  | CMPW0,1 |     |     |     |     |     |     |     |
| 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  $f_{sys} = 80$  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. These 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.

16.3.3 Conduction Control Circuit

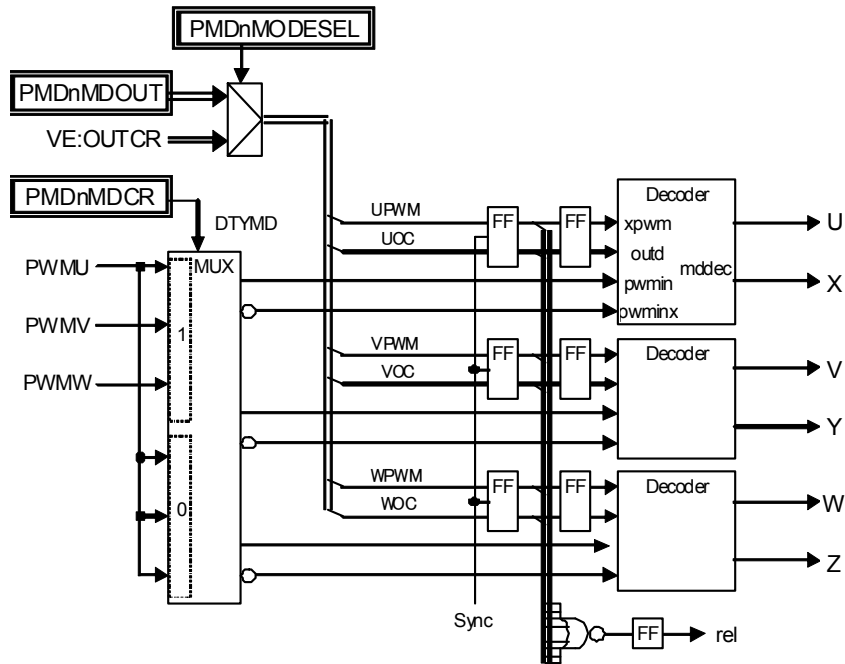


Figure 16-5 Conduction Control Circuit

The conduction control circuit performs output port control according to the settings made in the PMD output register (MDOUT). The MDOUT 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 16-3 shows a summary of port outputs according to port output settings in the MDOUT register and polarity settings in the MDCR register.

## 16.3.3.1 PMD Output Setting Register (PMDnMDPOT)

(PMD0:0x4005 042C, PMD1:0x4005 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   |

&lt;PSYNCS&gt;: 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.

&lt;POLL&gt;: 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.

&lt;POLH&gt;: 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.

16.3.3.2 PMD Output Control Register (PMDnMDOUT)

(PMD0:0x4005 0428, PMD1:0x4005 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 16-3 below.)

\* To load the second buffer of MDOUT with a value updated via the bus, select the bus mode (default) by setting MODESEL<MDESEL> 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.

OPMDnMDCR<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         | Bits 4, 2, 0 | 0: H/L output                                |                    | 1: PWM output      |                    |
| Upper phase          | Lower phase  | 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           | Bits 4,2,0  | 0: H/L output                                |                    | 1: PWM output      |                    |
| Upper phase          | Lower phase | 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                |

OPMDnMDCR<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         | Bits 4, 2, 0 | 0: H/L output                                |                    | 1: PWM output      |                    |
| Upper phase          | Lower phase  | 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           | Bits 4,2,0  | 0: H/L output                                |                    | 1: PWM output      |                    |
| Upper phase          | Lower phase | 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 16-3 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 16-4 Register Settings for Center-off PWM

|      | Normal PWM<br>center on | U-Phase PWM<br>center off | V-Phase PWM<br>center off | W-Phase PWM<br>center off |
|------|-------------------------|---------------------------|---------------------------|---------------------------|
| 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                        |



### 16.3.4 Protection Control Circuit

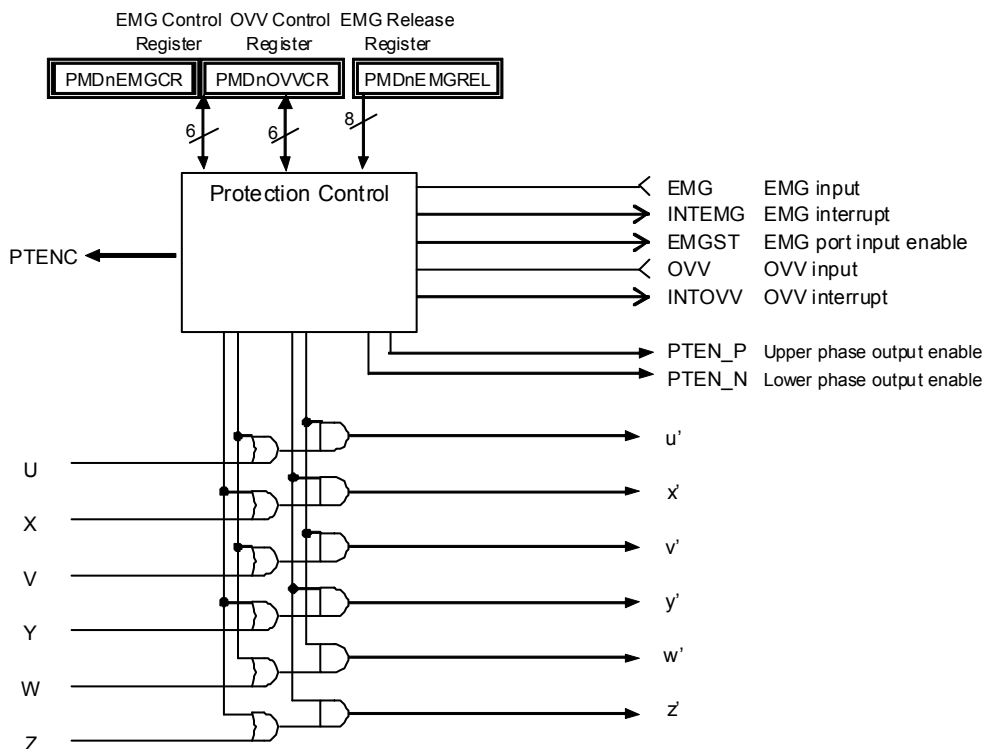


Figure 16-6 Protection Control Circuit

The protection control circuit consists of an EMG protection control circuit and an OVV protection control circuit.

#### 16.3.4.1 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.

## 16.3.4.2 EMG Release Register (PMDnEMGREL)

(PMD0:0x4005 0430, PMD1:0x4005 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   |

&lt;EMGREL&gt;: 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.

## 16.3.4.3 EMG Control Register (PMDnEMGCR)

(PMD0:0x4005 0434, PMD1:0x4005 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 |        | EMGISEL | 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     |

&lt;EMGEN&gt;: 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.

<EMGISEL>: EMG input select

0: Port input

1: Comparator output

This bit selects whether to use port input or comparator output as the EMG signal to be input to the protection circuit.

<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)

#### 16.3.4.4 EMG Status Register (PMDnEMGSTA)

(PMD0:0x4005 0438, PMD1:0x4005 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.

## 16.3.4.5 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 OVVCNT<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.

## 16.3.4.6 OVV Control Register (PMDnOVVCR)

(PMD0:0x4005 043C, PMD1:0x4005 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.)

$OVVCNT \times 16 / f_{sys}$  (resolution: 200[nsec] at 80 MHz)

\* OVVCNT is effective only when port input is selected as the OVV signal (<OVVISEL>=1).

### 16.3.4.7 OVV Status Register (PMDnOVVSTA) (PMD0:0x4005 0440, PMD1:0x4005 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.

16.3.5 Dead Time Circuit

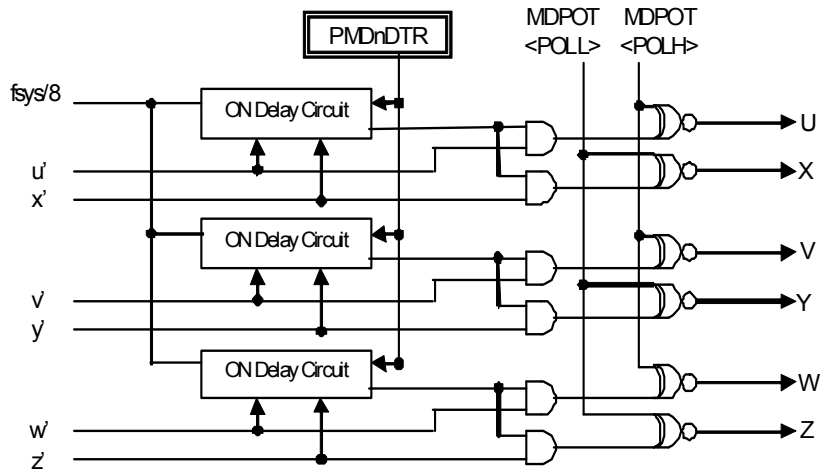


Figure 16-6 Dead Time Circuit

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>.

16.3.5.1 Dead Time Register (PMDnDTR)

(PMD0:0x4005 0444, PMD1:0x4005 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.

16.3.6 Sync Trigger Generation Circuit

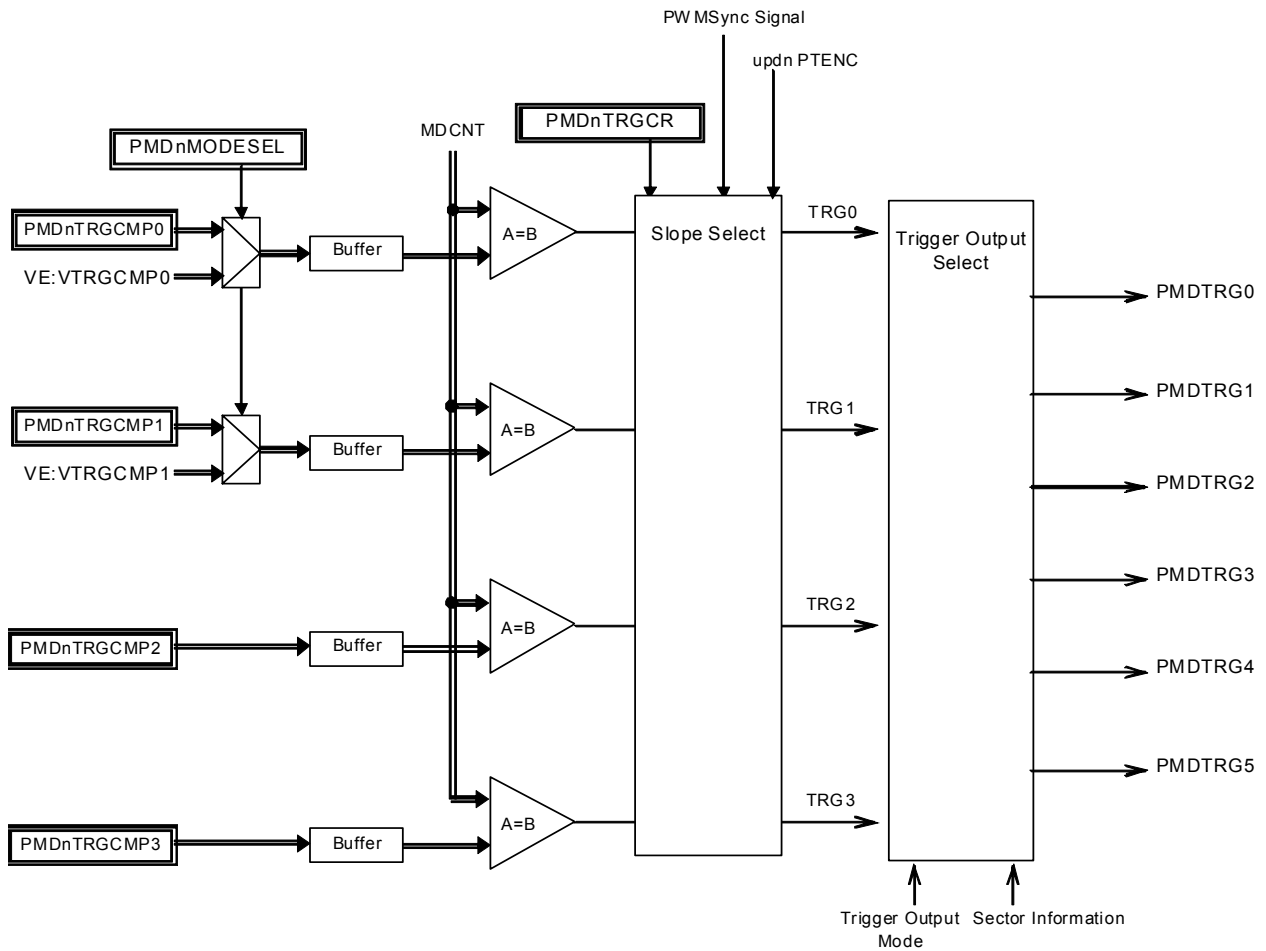


Figure 16-7 Sync Trigger Generation Circuit

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.



16.3.6.1 Trigger Compare Registers (PMDnTRGCMP0 ~3)

PMD0 (0x4005 0448-044B) , PMD1 (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   |

PMD0 (0x4005 044C-044F) , PMD1 (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   |

PMD0 (0x4005 0450-0453) , PMD1 (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   |

PMD0 (0x4005 0454-0458) , PMD1 (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<MSEL> 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 16-6 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                   |  |

16.3.6.2 Trigger Control Register (PMDnTRGCR)  
(PMD0:0x4005 0458, PMD1:0x4005 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<PWMMMD>=1 (triangular wave), one trigger output is made per period.

When the 1-shunt mode is used, the acceptable PMDTRG is as follows.

| VEFMODE0<br><IDMODE> | VEFMODE1<br><IDMODE> | PMD0TRGCR<br><TRG0MD> | PMD0TRGCR<br><TRG1MD> | PMD1TRGCR<br><TRG6MD> | PMD1TRGCR<br><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)  | -                     | -                     |
| -                    | 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.

### 16.3.6.3 Trigger Output Mode Setting Register (PMDnTRGMD) (PMD0:0x4005 045C, PMD1:0x4005 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 16-7 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 |                   |

### 16.3.6.4 Trigger Output Select Register (PMDnTRGSEL) (PMD0:0x4005 0460 , PMD1:0x4005 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 TRGCOMP0 value. (See Table 16-6.)

## 17. 12-Bit Analog-to-Digital Converters

The TMPM370 FY contains two 12-bit successive-approximation analog-to-digital converters (ADCs).

The ADC unit A (ADC A) has 15 analog inputs. Six inputs are able to use for shunt resistor currents of motor 0. Three of six inputs are connected with operational amp/ comparator outputs. Thus twelve inputs can use for external input.

The ADC unit B (ADC B) has 17 analog inputs. Six inputs are able to use for shunt resistor currents of motor 0. And two inputs are able to use for shut resistor currents of motor 1. four of eight inputs are connected with operational amp/ comparator outputs. Thus thirteen inputs can use for external input.

Twenty two external analog input pins (AINA0 to AINA8, AINA9/AINB0, AINA10/AINB1, AINA11/AINB2, AINB3 to AINB12) can also be used as input/output ports.

### Functions and features

- (1) It can select analog input and start AD conversion when receiving trigger signal from PMD or TMRB(interrupt).
- (2) It can select analog input, in the Software Trigger Program and the Constant Trigger Program.
- (3) The ADCs has twelve register for AD conversion result.
- (4) The ADCs generate interrupt signal at the end of the program which was started by PMD trigger and TMRB trigger.
- (5) The ADCs generate interrupt signal at the end of the program which are the Software Trigger Program and the Constant Trigger Program.
- (6) The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

### 17.1 Block Diagram

The following shows a block diagram of the ADCs.

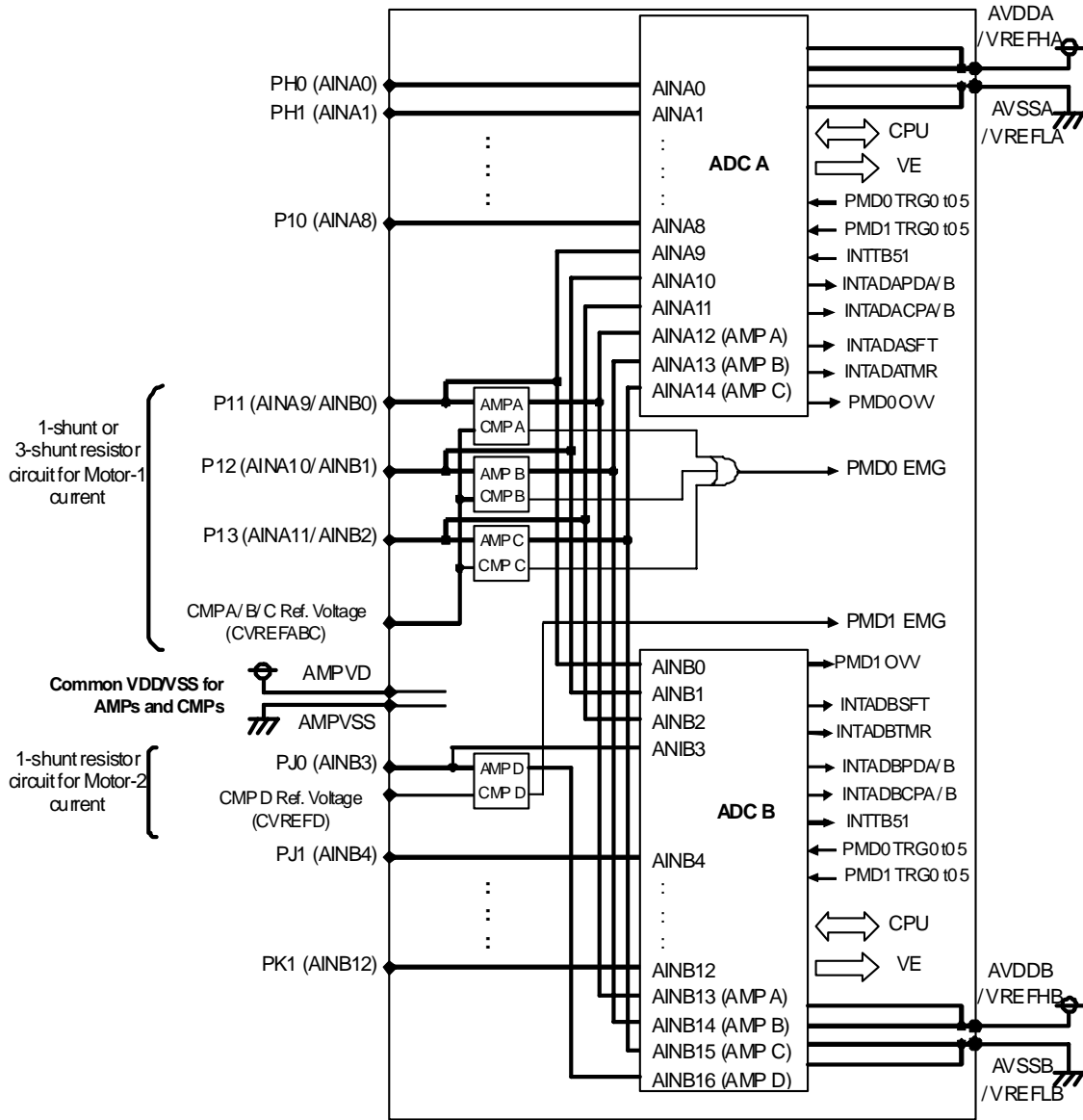


Fig 17.1 AD converters Block Diagram

## 17.2 List of Registers

The ADCs have the following registers.

Table 17.1 ADC Registers

| Address(Unit A) | Address(Unit B) | Register Name  | Description                                   |
|-----------------|-----------------|----------------|---|
| 4003_00_00      | 4003_02_00      | ADxCLK         | ADC Clock Setting Register                    |
| 4003_00_04      | 4003_02_04      | ADxMOD0        | ADC Mode Setting Register 0                   |
| 4003_00_08      | 4003_02_08      | ADxMOD1        | ADC Mode Setting Register 1                   |
| 4003_00_0C      | 4003_02_0C      | ADxMOD2        | ADC Mode Setting Register 2                   |
| 4003_00_10      | 4003_02_10      | ADxCMPCR0      | Monitoring Setting Register 0                 |
| 4003_00_14      | 4003_02_14      | ADxCMPCR1      | Monitoring Setting Register 1                 |
| 4003_00_18      | 4003_02_18      | ADxCMP0        | AD Conversion Result Compare Register 0       |
| 4003_00_1C      | 4003_02_1C      | ADxCMP1        | AD Conversion Result Compare Register 1       |
| 4003_00_20      | 4003_02_20      | ADxREG0        | AD Conversion Result Register 0               |
| 4003_00_24      | 4003_02_24      | ADxREG1        | AD Conversion Result Register 1               |
| 4003_00_28      | 4003_02_28      | ADxREG2        | AD Conversion Result Register 2               |
| 4003_00_2C      | 4003_02_2C      | ADxREG3        | AD Conversion Result Register 3               |
| 4003_00_30      | 4003_02_30      | ADxREG4        | AD Conversion Result Register 4               |
| 4003_00_34      | 4003_02_34      | ADxREG5        | AD Conversion Result Register 5               |
| 4003_00_38      | 4003_02_38      | ADxREG6        | AD Conversion Result Register 6               |
| 4003_00_3C      | 4003_02_3C      | ADxREG7        | AD Conversion Result Register 7               |
| 4003_00_40      | 4003_02_40      | ADxREG8        | AD Conversion Result Register 8               |
| 4003_00_44      | 4003_02_44      | ADxREG9        | AD Conversion Result Register 9               |
| 4003_00_48      | 4003_02_48      | ADxREG10       | AD Conversion Result Register 10              |
| 4003_00_4C      | 4003_02_4C      | ADxREG11       | AD Conversion Result Register 11              |
| 4003_00_50      | 4003_02_50      | ADxPSEL0       | PMD Trigger Program Number Select Register 0  |
| 4003_00_54      | 4003_02_54      | ADxPSEL1       | PMD Trigger Program Number Select Register 1  |
| 4003_00_58      | 4003_02_58      | ADxPSEL2       | PMD Trigger Program Number Select Register 2  |
| 4003_00_5C      | 4003_02_5C      | ADxPSEL3       | PMD Trigger Program Number Select Register 3  |
| 4003_00_60      | 4003_02_60      | ADxPSEL4       | PMD Trigger Program Number Select Register 4  |
| 4003_00_64      | 4003_02_64      | ADxPSEL5       | PMD Trigger Program Number Select Register 5  |
| 4003_00_68      | 4003_02_68      | ADxPSEL6       | PMD Trigger Program Number Select Register 6  |
| 4003_00_6C      | 4003_02_6C      | ADxPSEL7       | PMD Trigger Program Number Select Register 7  |
| 4003_00_70      | 4003_02_70      | ADxPSEL8       | PMD Trigger Program Number Select Register 8  |
| 4003_00_74      | 4003_02_74      | ADxPSEL9       | PMD Trigger Program Number Select Register 0  |
| 4003_00_78      | 4003_02_78      | ADxPSEL10      | PMD Trigger Program Number Select Register 10 |
| 4003_00_7C      | 4003_02_7C      | ADxPSEL11      | PMD Trigger Program Number Select Register 11 |
| 4003_00_80      | 4003_02_80      | ADxPINTS0      | PMD Trigger Interrupt Select Register 0       |
| 4003_00_84      | 4003_02_84      | ADxPINTS1      | PMD Trigger Interrupt Select Register 1       |
| 4003_00_88      | 4003_02_88      | ADxPINTS2      | PMD Trigger Interrupt Select Register 2       |
| 4003_00_8C      | 4003_02_8C      | ADxPINTS3      | PMD Trigger Interrupt Select Register 3       |
| 4003_00_90      | 4003_02_90      | ADxPINTS4      | PMD Trigger Interrupt Select Register 4       |
| 4003_00_94      | 4003_02_94      | ADxPINTS5      | PMD Trigger Interrupt Select Register 5       |
| 4003_00_98      | 4003_02_98      | ADxPSET0_REG   | PMD Trigger Program Register 0                |
| 4003_00_9C      | 4003_02_9C      | ADxPSET1_REG   | PMD Trigger Program Register 1                |
| 4003_00_A0      | 4003_02_A0      | ADxPSET2_REG   | PMD Trigger Program Register 2                |
| 4003_00_A4      | 4003_02_A4      | ADxPSET3_REG   | PMD Trigger Program Register 3                |
| 4003_00_A8      | 4003_02_A8      | ADxPSET4_REG   | PMD Trigger Program Register 4                |
| 4003_00_AC      | 4003_02_AC      | ADxPSET5_REG   | PMD Trigger Program Register 5                |
| 4003_00_B0      | 4003_02_B0      | ADxTSET_REG03  | Timer Trigger Program Registers 0 to 3        |
| 4003_00_B4      | 4003_02_B4      | ADxTSET_REG47  | Timer Trigger Program Registers 4 to 7        |
| 4003_00_B8      | 4003_02_B8      | ADxTSET_REG811 | Timer Trigger Program Registers 8 to 11       |
| 4003_00_BC      | 4003_02_BC      | ADxSSET_REG03  | Software Trigger Program Registers 0 to 3     |
| 4003_00_C0      | 4003_02_C0      | ADxSSET_REG47  | Software Trigger Program Registers 4 to 7     |
| 4003_00_C4      | 4003_02_C4      | ADxSSET_REG811 | Software Trigger Program Registers 8 to 11    |



| Address(Unit A) | Address(Unit B) | Register Name  | Description                                   |
|-----------------|-----------------|----------------|---|
| 4003_00_C8      | 4003_02_C8      | ADxASET_REG03  | Constant Conversion Program Registers 0 to 3  |
| 4003_00_CC      | 4003_02_CC      | ADxASET_REG47  | Constant Conversion Program Registers 4 to 7  |
| 4003_00_D0      | 4003_02_D0      | ADxASET_REG811 | Constant Conversion Program Registers 8 to 11 |
| 4003_00_D4      | 4003_02_D4      | ADxMOD3        | ADC Mode Setting Register 3                   |

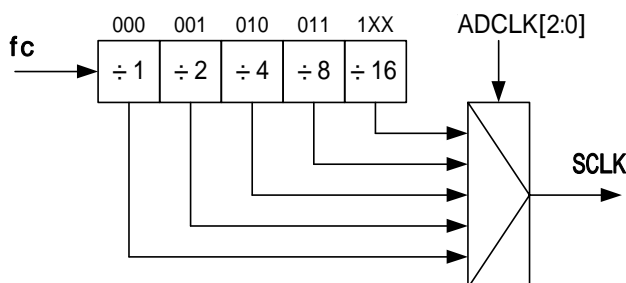
(Note) x= A or B : ADC Unit

### 17.3 Register Descriptions

AD conversion is performed at the clock frequency selected in the ADC Clock Setting Register.

#### 17.3.1 ADC Clock Setting Register (ADxCLK)

|                       |                   |              |      |      |      |   |        |        |   |
|-----------------------|-------------------|--------------|------|------|------|---|--------|--------|---|
| ADACLK<br>0x4003_0000 |                   | 7            | 6    | 5    | 4    | 3   | 2      | 1      | 0 |
| Bit symbol            | -                 | TSH3         | TSH2 | TSH1 | TSH0 | ADCLK2  | ADCLK1 | ADCLK0 |   |
| Read/Write            | R                 | R/W          |      |      |      | R/W   |        |        |   |
| ADBCLK<br>0x4003_0200 | After reset       | 0            | 1011 |      |      |   | 000    |        |   |
| Function              | Always read as 0. | Write "1001" |      |      |      | AD prescaler output (SCLK) select<br>000: fc (Note1) 001: fc/2<br>010: fc/4 011: fc/8<br>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.

#### 17.3.2 Mode Setting Registers

The ADC Mode Setting Registers (ADxMOD0, ADxMOD1, ADxMOD2 and ADxMOD3) are used to select how AD conversion is started. (x=A, B : ADC unit)

|                        |                   |   |   |   |   |   |   |                                |  |
|------------------------|-------------------|---|---|---|---|---|---|--------------------------------|--|
| ADxMOD0                |                   |   |   |   |   |   |   |                                |  |
| ADAMOD0<br>0x4003_0004 |                   | 7 | 6 | 5 | 4 | 3 | 2 | 1                              | 0  |
| Bit symbol             | -                 |   |   |   |   |   |   | DACON                          | ADSS   |
| Read/Write             | R                 | R |   |   |   |   |   | R/W                            | W  |
| ADBMOD0<br>0x4003_0204 | After reset       | 0 |   |   |   |   |   | 0                              | 0  |
| Function               | Always read as 0. |   |   |   |   |   |   | DAC control<br>0: off<br>1: On | Software triggered conversion<br>0: Don't care<br>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.

ADxMOD1

|                         |  |                   |   |   |   |   |   |   |   |   |
|-------------------------|--|-------------------|---|---|---|---|---|---|---|---|
| ADAMOD1<br>0xc4003_0008 |  | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0   |   |
| Bit symbol              | ADEN   | -                 |   |   |   |   |   |   | ADAS  |   |
| Read/Write              | R/W  | R                 |   |   |   |   |   |   | R/W   |   |
| ADBMOD1<br>0x4003_0208  | After reset                                      | 0                 | 0 |   |   |   |   |   |   | 0 |
| Function                | AD conversion control<br>0: Disable<br>1: Enable | Always read as 0. |   |   |   |   |   |   | Constant AD conversion control<br>0: Disable<br>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.

ADxMOD2

|                        |             |                   |   |   |   |   |   |   |   |   |
|------------------------|-------------|-------------------|---|---|---|---|---|---|---|---|
| ADAMOD2<br>0x4003_000C |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0   |   |
| Bit symbol             |             | -                 |   |   |   |   |   |   | ADSFN   | ADBFN   |
| Read/Write             |             | R                 |   |   |   |   |   |   | R   | R   |
| ADBMOD2<br>0x4003_020C | After reset | 0                 |   |   |   |   |   |   | 0   | 0   |
| Function               |             | Always read as 0. |   |   |   |   |   |   | Software conversion busy flag<br>0: Conversion completed<br>1: Conversion in progress | AD conversion busy flag<br>0: Conversion not in progress<br>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”.

ADxMOD3

|                        |             |           |               |               |               |           |           |           |   |
|------------------------|-------------|-----------|---------------|---------------|---------------|-----------|-----------|-----------|---|
| ADAMOD3<br>0x4003_00D4 |             | 7         | 6             | 5             | 4             | 3         | 2         | 1         | 0 |
| Bit symbol             | -           | -         | <b>PMODE2</b> | <b>PMODE1</b> | <b>PMODE0</b> | -         | -         | -         |   |
| Read/Write             | R/W         | R/W       | R/W           | R/W           | R/W           | R/W       | R/W       | R/W       |   |
| ADBMOD3<br>0x4003_02D4 | After reset | 0         | 1             | 0             | 1             | 1         | 0         | 0         | 0 |
| Function               | Write “0”   | Write “1” | Write “100”   |               |               | Write “0” | Write “0” | Write “0” |   |
|                        | 15          | 14        | 13            | 12            | 11            | 10        | 9         | 8         |   |
| Bit symbol             | -           | -         | -             | -             | -             | -         | -         | -         |   |
| 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         | 0         | 1         |   |
| Function               | Write “0”   | Write “0” | Write “0”     | Write “0”     | Write “0”     | Write “1” | Write “0” | Write “1” |   |

Note : <PMODE[2:0]> must be set to “100”. And do not change other bits in ADxMOD3 register.

### 17.3.3 Monitoring Setting Registers

The ADCs have the AD conversion result monitoring function.

#### 17.3.3.1 ADxCMPCR0, ADxCMPCR1

The ADxCMPCR0 and ADxCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.

After fixing the conversion result, the interrupt signal (INTADxCPA, INTADxCPB) is generated.

(x=A , B : ADC unit)

#### ADxCMPCR0

|                          |             |  |                   |                   |   |  |          |          |          |
|--------------------------|-------------|--|-------------------|-------------------|---|--|----------|----------|----------|
| ADACMPCR0<br>0x4003_0010 |             | 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      |
| ADBCMPCR0<br>0x4003_0210 | After reset | 0  | 0                 | 0                 | 0   | 0  | 0        | 0        | 0        |
|                          | Function    | Monitoring function<br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Comparison condition<br>0: Larger than or equal to compare register<br>1: Smaller than or equal to compare register | AD conversion result register to be compared<br>0000: ADREG0 0100: ADREG4 1000: ADREG8<br>0001: ADREG1 0101: ADREG5 1001: ADREG9<br>0010: ADREG2 0110: ADREG6 1010: ADREG10<br>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<br>0000: After every comparison<br>0001: After two comparisons<br>.<br>.<br>1111: After 16 comparisons   |          |          |          |

#### ADxCMPCR1

|                          |             |  |                   |                   |   |  |          |          |          |
|--------------------------|-------------|--|-------------------|-------------------|---|--|----------|----------|----------|
| ADACMPCR1<br>0x4003_0014 |             | 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      |
| ADBCMPCR1<br>0x4003_0214 | After reset | 0  | 0                 | 0                 | 0   | 0  | 0        | 0        | 0        |
|                          | Function    | Monitoring function<br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Comparison condition<br>0: Larger than or equal to compare register<br>1: Smaller than or equal to compare register | AD conversion result register to be compared<br>0000: ADREG0 0100: ADREG4 1000: ADREG8<br>0001: ADREG1 0101: ADREG5 1001: ADREG9<br>0010: ADREG2 0110: ADREG6 1010: ADREG10<br>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<br>0000: After every comparison<br>0001: After two comparisons<br>.<br>.<br>1111: After 16 comparisons   |          |          |          |

### 17.3.3.2 Conversion Result Compare Register

The ADxCMP0 and ADxCMP1 registers specify the value to be compared with an AD conversion result. The upper 12 bits (bits 4 to 15) are used. (x=A , B : ADC unit)

#### ADxCMP0

|                        |             |   |          |          |          |                   |          |          |          |
|------------------------|-------------|---|----------|----------|----------|-------------------|----------|----------|----------|
| ADACMP0<br>0x4003_0018 |             | 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        |
| ADBCMP0<br>0x4003_0218 | 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 |          |          |          |                   |          |          |          |

#### ADxCMP1

|                        |             |   |          |          |          |                   |          |          |          |
|------------------------|-------------|---|----------|----------|----------|-------------------|----------|----------|----------|
| ADACMP1<br>0x4003_001C |             | 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        |
| ADBCMP1<br>0x4003_021C | 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 |          |          |          |                   |          |          |          |

### 17.3.4 AD Conversion Result Registers

#### 17.3.4.1 ADxREG0 to ADxREG11

The ADxREGn (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 ADxREGn register and is cleared when the low-order byte of ADxREGn 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 ADxREGn is read and is cleared when the low-order byte of ADxREGn is read.

There are twelve ADxREGn registers, which are all functionally equivalent. Only the ADREG0 register is explained here. (x=A , B : ADC unit)

AD Conversion Result Register (ADxREG0)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG0<br>0x4003_0020 |             | 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  |
| ADBREG0<br>0x4003_0220 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG1)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG1<br>0x4003_0024 |             | 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  |
| ADBREG0<br>0x4003_0224 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG2)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG2<br>0x4003_0028 |             | 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  |
| ADBREG2<br>0x4003_0228 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG3)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG3<br>0x4003_002C |             | 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  |
| ADBREG0<br>0x4003_022C | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG4)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG4<br>0x4003_0030 |             | 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  |
| ADBREG4<br>0x4003_0230 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG5)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG5<br>0x4003_0034 |             | 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  |
| ADBREG5<br>0x4003_0234 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG6)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG6<br>0x4003_0038 |             | 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  |
| ADBREG6<br>0x4003_0238 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG7)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG7<br>0x4003_003C |             | 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  |
| ADBREG7<br>0x4003_023C | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG8)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG8<br>0x4003_0040 |             | 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  |
| ADBREG8<br>0x4003_0240 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG9)

|                        |             |   |        |        |        |                   |                   |   |  |
|------------------------|-------------|---|--------|--------|--------|-------------------|-------------------|---|--|
| ADAREG9<br>0x4003_0044 |             | 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  |
| ADBREG9<br>0x4003_0244 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG10)

|                         |             |   |         |         |         |                   |                   |   |  |
|-------------------------|-------------|---|---------|---------|---------|-------------------|-------------------|---|--|
| ADAREG10<br>0x4003_0048 |             | 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  |
| ADBREG10<br>0x4003_0248 | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 (ADxREG11)

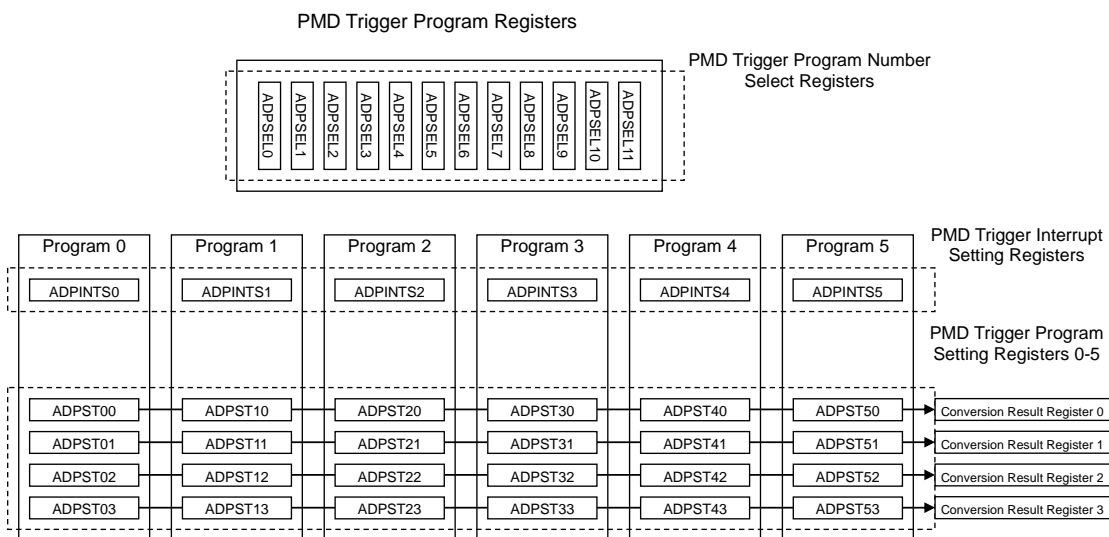
|                         |             |   |         |         |         |                   |                   |   |  |
|-------------------------|-------------|---|---------|---------|---------|-------------------|-------------------|---|--|
| ADAREG11<br>0x4003_004C |             | 7                                       | 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  |
| ADBREG11<br>0x4003_024C | 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<br>0: No overrun occurred<br>1: Overrun occurred | AD conversion result store flag<br>0: No result stored<br>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 |         |         |         |                   |                   |   |  |

### 17.3.5 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 each of twelve triggers generated by the PMD, to select the interrupt to be generated upon completion of the program and to select the AIN input to be used.

The PMD trigger program registers include three types of registers: PMD Trigger Program Number Select Register (ADxPSELp), PMD Trigger Interrupt Select Register (ADxPINTSn) and PMD Trigger Program Setting Register (ADxPSETnm). (x=A, B : ADC unit , p=0 to 11, n=0 to 5, m=0 to 3)



The PMD Trigger Program Number Select Register (ADxPSELn) specifies the program to be started by each of twelve AD conversion start signals corresponding to twelve triggers generated by the PMD. Programs 0 to 5 are available.

The PMD Trigger Interrupt Select Register (ADxPINTSn) selects the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

The PMD Trigger Program Setting Register (ADxPSETnm) 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 ADxPSETn0 to ADxPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADxREG0 to ADxREG3).

### 17.3.5.1 PMD Trigger Program Number Select Registers

The PMD Trigger Program Number Select Registers (ADxPSEL0 to ADxPSEL11) select the program to be started (from among programs 0 to 5) by each of trigger inputs PMD0 to PMD11.

#### PMD Trigger Program Number Select Register 0 (ADxPSEL0)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL0<br>0x4003_0050  |             | 7   | 6                 | 5                 | 4                 | 3                 | 2   | 1      | 0      |
|                          | Bit symbol  | PENS0   |                   |                   |                   |                   | PMDS02  | PMDS01 | PMDS00 |
|                          | Read/Write  | R/W   | R                 | R                 | R                 | R                 | R/W   | R/W    | R/W    |
| ADBPSSEL0<br>0x4003_0250 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 0 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

#### PMD Trigger Program Number Select Register 1 (ADxPSEL1)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL1<br>0x4003_0054  |             | 7   | 6                 | 5                 | 4                 | 3                 | 2   | 1      | 0      |
|                          | Bit symbol  | PENS1   |                   |                   |                   |                   | PMDS12  | PMDS11 | PMDS10 |
|                          | Read/Write  | R/W   | R                 | R                 | R                 | R                 | R/W   | R/W    | R/W    |
| ADBPSSEL1<br>0x4003_0254 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 1 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

#### PMD Trigger Program Number Select Register 2 (ADxPSEL2)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL2<br>0x4003_0058  |             | 7   | 6                 | 5                 | 4                 | 3                 | 2   | 1      | 0      |
|                          | Bit symbol  | PENS2   |                   |                   |                   |                   | PMDS22  | PMDS21 | PMDS20 |
|                          | Read/Write  | R/W   | R                 | R                 | R                 | R                 | R/W   | R/W    | R/W    |
| ADBPSSEL2<br>0x4003_0258 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 2 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

#### PMD Trigger Program Number Select Register 3 (ADxPSEL3)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL3<br>0x4003_005C  |             | 7   | 6                 | 5                 | 4                 | 3                 | 2   | 1      | 0      |
|                          | Bit symbol  | PENS3   |                   |                   |                   |                   | PMDS32  | PMDS31 | PMDS30 |
|                          | Read/Write  | R/W   | R                 | R                 | R                 | R                 | R/W   | R/W    | R/W    |
| ADBPSSEL3<br>0x4003_025C | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 3 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 4 (ADxPSEL4)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL4<br>0x4003_0060  |             | 7   | 6                 | 5                 | 4                 | 3                 | 2   | 1      | 0      |
|                          | Bit symbol  | PENS4   |                   |                   |                   |                   | PMDS42  | PMDS41 | PMDS40 |
|                          | Read/Write  | R/W   | R                 | R                 | R                 | R                 | R/W   | R/W    | R/W    |
| ADBPSSEL4<br>0x4003_0260 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 4 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 5 (ADxPSEL5)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL5<br>0x4003_0064  |             | 7   | 6                 | 5                 | 4                 | 3                 | 2   | 1      | 0      |
|                          | Bit symbol  | PENS5   |                   |                   |                   |                   | PMDS52  | PMDS51 | PMDS50 |
|                          | Read/Write  | R/W   | R                 | R                 | R                 | R                 | R/W   | R/W    | R/W    |
| ADBPSSEL5<br>0x4003_0264 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 5 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 6 (ADxPSEL6)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL6<br>0x4003_0068  |             | 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    |
| ADBPSSEL6<br>0x4003_0268 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 6 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 7 (ADxPSEL7)

|                          |             |   |                   |                   |                   |                   |   |        |        |
|--------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL7<br>0x4003_006C  |             | 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    |
| ADBPSSEL7<br>0x4003_026C | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                          | Function    | PMD Trigger Program Select Register 7 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 8 (ADxPSEL8)

|                         |             |   |                   |                   |                   |                   |   |        |        |
|-------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL8<br>0x4003_0070 |             | 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    |
| ADBPSEL8<br>0x4003_0270 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                         | Function    | PMD Trigger Program Select Register 8 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 9 (ADxPSEL9)

|                         |             |   |                   |                   |                   |                   |   |        |        |
|-------------------------|-------------|---|-------------------|-------------------|-------------------|-------------------|---|--------|--------|
| ADAPSEL9<br>0x4003_0074 |             | 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    |
| ADBPSEL9<br>0x4003_0274 | After reset | 0   | 0                 | 0                 | 0                 | 0                 | 0   | 0      | 0      |
|                         | Function    | PMD Trigger Program Select Register 9 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |        |        |

PMD Trigger Program Number Select Register 10 (ADxPSEL10)

|                          |             |  |                   |                   |                   |                   |   |         |         |
|--------------------------|-------------|--|-------------------|-------------------|-------------------|-------------------|---|---------|---------|
| ADAPSEL10<br>0x4003_0078 |             | 7  | 6                 | 5                 | 4                 | 3                 | 2   | 1       | 0       |
|                          | Bit symbol  | PENS10   |                   |                   |                   |                   | PMDS102   | PMDS101 | PMDS100 |
|                          | Read/Write  | R/W  | R                 | R                 | R                 | R                 | R/W   | R/W     | R/W     |
| ADBPSEL10<br>0x4003_0278 | After reset | 0  | 0                 | 0                 | 0                 | 0                 | 0   | 0       | 0       |
|                          | Function    | PMD Trigger Program Select Register 10 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010 Program 2    110,111: Reserved<br>011: Program 3 |         |         |

PMD Trigger Program Number Select Register 11 (ADxPSEL11)

|                          |             |  |                   |                   |                   |                   |   |         |         |
|--------------------------|-------------|--|-------------------|-------------------|-------------------|-------------------|---|---------|---------|
| ADAPSEL11<br>0x4003_007C |             | 7  | 6                 | 5                 | 4                 | 3                 | 2   | 1       | 0       |
|                          | Bit symbol  | PENS11   |                   |                   |                   |                   | PMDS112   | PMDS111 | PMDS110 |
|                          | Read/Write  | R/W  | R                 | R                 | R                 | R                 | R/W   | R/W     | R/W     |
| ADBPSEL11<br>0x4003_027C | After reset | 0  | 0                 | 0                 | 0                 | 0                 | 0   | 0       | 0       |
|                          | Function    | PMD Trigger Program Select Register 11 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | Always read as 0. | Always read as 0. | Program number select<br><br>000: Program 0    100: Program 4<br>001: Program 1    101: Program 5<br>010: Program 2    110, 111: Reserved<br>011: Program 3 |         |         |

### 17.3.5.2 PMD Trigger Interrupt Select Registers

The PMD Trigger Interrupt Select Registers (ADPINTS0 to ADPINTS5) select the interrupt to be generated for each of programs 0 to 5.

#### ADxPINTS0 (for program 0)

|                          |             |                   |   |   |   |   |   |   |          |
|--------------------------|-------------|-------------------|---|---|---|---|---|---|----------|
|                          |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1   | 0        |
| ADAPINTS0<br>0x4003_0080 | Bit symbol  |                   |   |   |   |   |   | INTSEL01  | INTSEL00 |
|                          | Read/Write  | R                 |   |   |   |   |   | R/W   |          |
| ADBPINTS0<br>0x4003_0280 | After reset | 0                 |   |   |   |   |   | 0   |          |
|                          | Function    | Always read as 0. |   |   |   |   |   | Interrupt select<br>00: No interrupt output<br>01: INTADPDA<br>10: INTADPDB<br>11: Reserved |          |

Note: When <INTSEL> = 11, no interrupt is generated.

#### ADxPINTS1 (for program 1)

|                          |             |                   |   |   |   |   |   |   |          |
|--------------------------|-------------|-------------------|---|---|---|---|---|---|----------|
|                          |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1   | 0        |
| ADAPINTS1<br>0x4003_0084 | Bit symbol  |                   |   |   |   |   |   | INTSEL11  | INTSEL10 |
|                          | Read/Write  | R                 |   |   |   |   |   | R/W   |          |
| ADBPINTS1<br>0x4003_0284 | After reset | 0                 |   |   |   |   |   | 0   |          |
|                          | Function    | Always read as 0. |   |   |   |   |   | Interrupt select<br>00: No interrupt output<br>01: INTADPDA<br>10: INTADPDB<br>11: Reserved |          |

Note: When <INTSEL> = 11, no interrupt is generated.

#### ADxPINTS2 (for program 2)

|                          |             |                   |   |   |   |   |   |   |          |
|--------------------------|-------------|-------------------|---|---|---|---|---|---|----------|
|                          |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1   | 0        |
| ADAPINTS2<br>0x4003_0088 | Bit symbol  |                   |   |   |   |   |   | INTSEL21  | INTSEL20 |
|                          | Read/Write  | R                 |   |   |   |   |   | R/W   |          |
| ADBPINTS2<br>0x4003_0288 | After reset | 0                 |   |   |   |   |   | 0   |          |
|                          | Function    | Always read as 0. |   |   |   |   |   | Interrupt select<br>00: No interrupt output<br>01: INTADPDA<br>10: INTADPDB<br>11: Reserved |          |

Note: When <INTSEL> = 11, no interrupt is generated.

#### ADxPINTS3 (for program 3)

|                          |             |                   |   |   |   |   |   |   |          |
|--------------------------|-------------|-------------------|---|---|---|---|---|---|----------|
|                          |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1   | 0        |
| ADAPINTS3<br>0x4003_008C | Bit symbol  |                   |   |   |   |   |   | INTSEL31  | INTSEL30 |
|                          | Read/Write  | R                 |   |   |   |   |   | R/W   |          |
| ADBPINTS3<br>0x4003_028C | After reset | 0                 |   |   |   |   |   | 0   |          |
|                          | Function    | Always read as 0. |   |   |   |   |   | Interrupt select<br>00: No interrupt output<br>01: INTADPDA<br>10: INTADPDB<br>11: Reserved |          |

Note: When <INTSEL> = 11, no interrupt is generated.

ADxPINTS4 (for program 4)

|                          |             |                   |   |   |   |   |   |   |          |
|--------------------------|-------------|-------------------|---|---|---|---|---|---|----------|
| ADAPINTS4<br>0x4003_0090 |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1   | 0        |
|                          | Bit symbol  |                   |   |   |   |   |   | INTSEL41  | INTSEL40 |
|                          | Read/Write  | R                 |   |   |   |   |   | R/W   |          |
| ADBPINTS4<br>0x4003_0290 | After reset | 0                 |   |   |   |   |   | 0   |          |
|                          | Function    | Always read as 0. |   |   |   |   |   | Interrupt select<br>00: No interrupt output<br>01: INTADPDA<br>10: INTADPDB<br>11: Reserved |          |

Note: When <INTSEL> = 11, no interrupt is generated.

ADxPINTS5 (for program 5)

|                          |             |                   |   |   |   |   |   |   |          |
|--------------------------|-------------|-------------------|---|---|---|---|---|---|----------|
| ADAPINTS5<br>0x4003_0094 |             | 7                 | 6 | 5 | 4 | 3 | 2 | 1   | 0        |
|                          | Bit symbol  |                   |   |   |   |   |   | INTSEL51  | INTSEL50 |
|                          | Read/Write  | R                 |   |   |   |   |   | R/W   |          |
| ADBPINTS5<br>0x4003_0294 | After reset | 0                 |   |   |   |   |   | 0   |          |
|                          | Function    | Always read as 0. |   |   |   |   |   | Interrupt select<br>00: No interrupt output<br>01: INTADPDA<br>10: INTADPDB<br>11: Reserved |          |

Note: When <INTSEL> = 11, no interrupt is generated.



### 17.3.5.3 PMD Trigger Program Setting Registers 0 to 5

Each of the PMD Trigger Program Registers (ADxPSET0 to 5) are comprised of four registers. These four registers are used to select the AD conversion input pin (AINA0 to AINA14, AINB0 to AINB16) 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 <ENSP<sub>nm</sub>> to 1 enables the ADxPSET<sub>nm</sub> register. The <UVWIS<sub>nm</sub>> bits are used to select phase-U, phase-V or phase-W. The <AINSP<sub>nm</sub>> 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 (ADxPSET0)

ADAPSET0  
0x4003\_0098

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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET0</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET0</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET0</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET0</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET0</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET0</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET0</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET0</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

PMD Trigger Program Register 1 (ADxPSET1)

ADAPSET1  
0x4003\_009C

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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET1</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET1</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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   |          |          |          |          |
| Function    | REG1 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET1</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET1</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET1</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET1</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG3 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET1</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET1</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

PMD Trigger Program Register 2(ADxPSET2)

ADAPSET2  
0x4003\_00A0  
  
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<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET2</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |   |          | AIN select (ADC unit B) in case of <b>ADBPSET2</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET2</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |   |          | AIN select (ADC unit B) in case of <b>ADBPSET2</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET2</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |   |          | AIN select (ADC unit B) in case of <b>ADBPSET2</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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   |          | 0   |          |          |          |          |
| Function    | REG3<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET2</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |   |          | AIN select (ADC unit B) in case of <b>ADBPSET2</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

PMD Trigger Program Register 3(ADxPSET3)

ADAPSET3  
0x4003\_00A4

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<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET3</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET3</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG1 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET3</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET3</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG2 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET3</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET3</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG3 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET3</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET3</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

PMD Trigger Program Register 4(ADxPSET4)

ADAPSET4  
0x4003\_00A8

ADBPSET4  
0x4003\_02A8

|             |   |   |   |          |          |          |          |          |
|-------------|---|---|---|----------|----------|----------|----------|----------|
|             | 7   | 6   | 5   | 4        | 3        | 2        | 1        | 0        |
| Bit symbol  | ENSP40  | UVWIS401  | UVWIS400  | AINSP404 | AINSP403 | AINSP402 | AINSP401 | AINSP400 |
| Read/Write  | R/W   | R/W   |   | R/W      |          |          |          |          |
| After reset | 0   | 0   |   | 0        |          |          |          |          |
| Function    | REG0<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W | AIN select (ADC unit A) in case of <b>ADAPSET4</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |          |
|             |   |   | AIN select (ADC unit B) in case of <b>ADBPSET4</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W | AIN select (ADC unit A) in case of <b>ADAPSET4</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |          |
|             |   |   | AIN select (ADC unit B) in case of <b>ADBPSET4</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |          |
|             | 23  | 22  | 21  | 20       | 19       | 18       | 17       | 16       |
| Bit symbol  | ENSP42  | UVWIS421  | UVWIS420  | AINSP424 | AINSP423 | AINSP422 | AINSP421 | AINSP420 |
| Read/Write  | R/W   | R/W   |   | R/W      |          |          |          |          |
| After reset | 0   | 0   |   | 0        |          |          |          |          |
| Function    | REG2<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W | AIN select (ADC unit A) in case of <b>ADAPSET4</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |          |
|             |   |   | AIN select (ADC unit B) in case of <b>ADBPSET4</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br>enable<br><br>0: Disable<br>1: Enable | Phase select<br>(for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W | AIN select (ADC unit A) in case of <b>ADAPSET4</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |          |
|             |   |   | AIN select (ADC unit B) in case of <b>ADBPSET4</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

PMD Trigger Program Register 5(ADxPSET5)

ADAPSET5  
0x4003\_00AC

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  |          | 0   |          |          |          |          |
| Function    | REG0 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET5</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET5</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG1 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET5</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET5</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG2 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET5</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET5</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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  |          | 0   |          |          |          |          |
| Function    | REG3 enable<br><br>0: Disable<br>1: Enable | Phase select (for Vector Engine)<br><br>00: Not specified<br>01: U<br>10: V<br>11: W |          | AIN select (ADC unit A) in case of <b>ADAPSET5</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |  |  |          | AIN select (ADC unit B) in case of <b>ADBPSET5</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

### 17.3.6 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 ADxTSETm 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 : INTADxTMR is generated.

(x=A , B : ADC unit , m= 0 to 11, n= 0 to 4)



Timer Trigger Program Register 03 (ADxTSET03)

ADATSET03  
0x4003\_00B0  
  
ABDTSET03  
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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |         |         |         |         |

Timer Trigger Program Register 47 (ADxTSET47)

ADATSET47  
0x4003\_00B4

ABDTSET47  
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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ABDTSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |         |         |         |         |

Timer Trigger Program Register 811 (ADTSET811)

ADATSET811  
0x4003\_00B8

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<br><br>0: Disable<br>1: Enable  | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBTSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable  | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBTSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBTSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADATSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBTSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

### 17.3.7 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 ADxSSETn 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 :INTADxSFT is generated.

(x=A , B : ADC unit , n= 0 to 11, m= 0 to 4)

Software Trigger Program Register 03 (ADxSSET03)

ADASSET03  
0x4003\_00BC  
  
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                 | 0  |         |         |         |         |
| Function    | REG0 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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                 | 0  |         |         |         |         |
| Function    | REG1 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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                 | 0  |         |         |         |         |
| Function    | REG2 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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                 | 0  |         |         |         |         |
| Function    | REG3 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |         |         |         |         |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

Software Trigger Program Register 47 (ADxSSET47)

ADASSET47  
0x4003\_00C0  
  
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                 | 0  |         |         |         |         |
| Function    | REG4 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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                 | 0  |         |         |         |         |
| Function    | REG5 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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                 | 0  |         |         |         |         |
| Function    | REG6 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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                 | 0  |         |         |         |         |
| Function    | REG7 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |         |         |         |         |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

Software Trigger Program Register 811 (ADxSSET811)

|                            |             |   |                   |                   |   |          |          |          |          |
|----------------------------|-------------|---|-------------------|-------------------|---|----------|----------|----------|----------|
| ADASSET811<br>0x4003_00C4  |             | 7   | 6                 | 5                 | 4   | 3        | 2        | 1        | 0        |
|                            | Bit symbol  | ENSS8                                       | -                 | -                 | AINSS84   | AINSS83  | AINSS82  | AINSS81  | AINSS80  |
|                            | Read/Write  | R/W   | R                 | R                 | R/W   |          |          |          |          |
| ADBSSSET811<br>0x4003_02C4 | After reset | 0   | 0                 | 0                 | 0   |          |          |          |          |
|                            | Function    | REG8 enable<br><br>0: Disable<br>1: Enable  | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|                            |             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |
|                            | Bit symbol  | ENSS9                                       | -                 | -                 | AINSS94   | AINSS93  | AINSS92  | AINSS91  | AINSS90  |
|                            | Read/Write  | R/W   | R                 | R                 | R/W   |          |          |          |          |
|                            | After reset | 0   | 0                 | 0                 | 0   |          |          |          |          |
|                            | Function    | REG9 enable<br><br>0: Disable<br>1: Enable  | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|                            |             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |
|                            | Bit symbol  | ENSS10                                      | -                 | -                 | AINSS104  | AINSS103 | AINSS102 | AINSS101 | AINSS100 |
|                            | Read/Write  | R/W   | R                 | R                 | R/W   |          |          |          |          |
|                            | After reset | 0   | 0                 | 0                 | 0   |          |          |          |          |
|                            | Function    | REG10 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|                            |             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |
|                            | Bit symbol  | ENSS11                                      | -                 | -                 | AINSS114  | AINSS113 | AINSS112 | AINSS111 | AINSS110 |
|                            | Read/Write  | R/W   | R                 | R                 | R/W   |          |          |          |          |
|                            | After reset | 0   | 0                 | 0                 | 0   |          |          |          |          |
|                            | Function    | REG11 enable<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASSET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|                            |             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADASSET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

### 17.3.8 Constant Conversion Program Registers

The ADCs allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSA<sub>m</sub>> to “1” enables the ADxASET<sub>m</sub> register. The <AINSA<sub>n</sub>m> 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.

(x=A, B : ADC unit , m= 0 to 11, n= 0 to 4)



Constant Conversion Program Register 03 (ADxASET03)

ADAASET0  
0x4003\_00C8  
  
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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET03</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET03</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |         |         |         |         |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

Constant Conversion Program Register 47 (ADxASET47)

|                          |             |  |                   |                   |   |         |         |         |         |
|--------------------------|-------------|--|-------------------|-------------------|---|---------|---------|---------|---------|
| ADASET47<br>0x4003_00CC  |             | 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|                          |             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved) |         |         |         |         |
| ADBASET47<br>0x4003_02CC |             | 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|                          |             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|                          |             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADASET47</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |         |         |         |         |
|                          |             |  |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET47</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved) |         |         |         |         |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

Constant Conversion Program Register 811 (ADxASET811)

ADAASET8  
0x4003\_00D0  
  
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<br><br>0: Disable<br>1: Enable  | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable  | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 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<br><br>0: Disable<br>1: Enable | Always read as 0. | Always read as 0. | AIN select (ADC unit A) in case of <b>ADAASET811</b><br>00000: AINA0<br>00001: AINA1<br>:<br>01110: AINA14 (01111 to 11111: Reserved) |          |          |          |          |
|             |   |                   |                   | AIN select (ADC unit B) in case of <b>ADBASET811</b><br>00000: AINB0<br>00001: AINB1<br>:<br>10000: AIN16 (10001 to 11111: Reserved)  |          |          |          |          |

Note: When AIN24 to AIN 31 (set value: 11000 to 11111) are set in the AIN select field, AIN0 is selected.

## 17.4 Operation Descriptions

### 17.4.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the AVDD5A and AVSSA pins are used in ADC A and the AVDD5B and AVSSB pins are used in ADC B. There are no registers for controlling current between AVDD5x and AVSSx. Inputs to these pins are fixed. (x=A,B:ADC unit)

The internal amplifiers and comparators share the power supply and GND, which are connected to AMPVDD and AMPVSS respectively.

**(Note 1) During AD conversion, do not change the output data of port H/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.**

### 17.4.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD trigger (See “17.3.5 PMD Trigger Program Setting Registers.”)
- Timer trigger (TMRB5) (See “17.3.6 Timer Trigger Program Setting Registers.”)
- Software trigger (See “17.3.7 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.

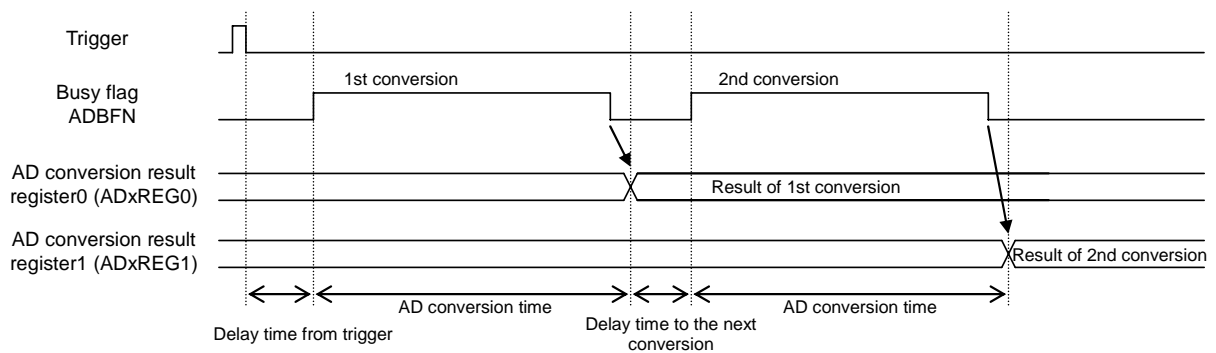


Figure Timing chart of A/D conversion

Table A/D conversion time (SCLK=40MHz)

|  |                          | fsys=80MHz |       | fsys=40MHz |       |
|--|--------------------------|------------|-------|------------|-------|
|  |                          | 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.

### 17.4.3 AD Conversion Monitoring Function

The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function, set ADxCMPCR0<CMP0EN> or ADxCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by ADxCMCR<ADBIG0>, the interrupt (INTADxCPA for ADxCMPCR0, INTADxCPB for ADxCMPCR1) is generated. The comparison is executed at the timing of storing the conversion result into the register.

(x=A, B : ADC unit)

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 (OVRs) is set.

### 17.5 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

#### 17.5.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADxSSET03, ADxSSET47 and ADxSSET811. If the ADxMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.

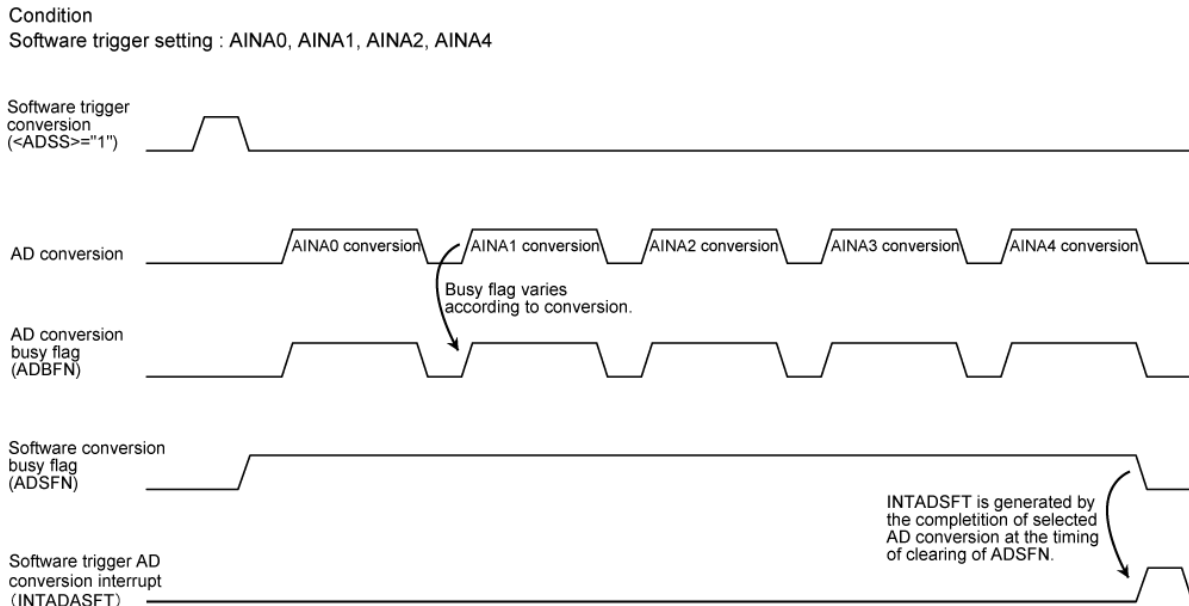


Fig 17-3 Software trigger AD conversion

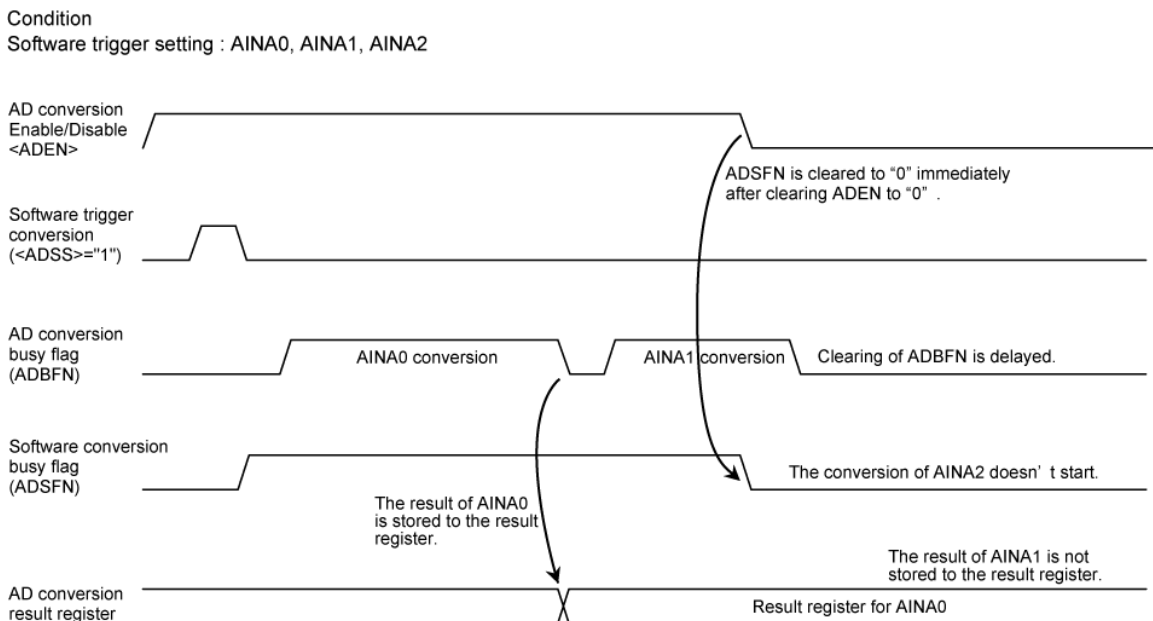


Fig 17-4 Writing "0" to ADEN during the software trigger AD conversion

### 17.5.2 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 : AINA0

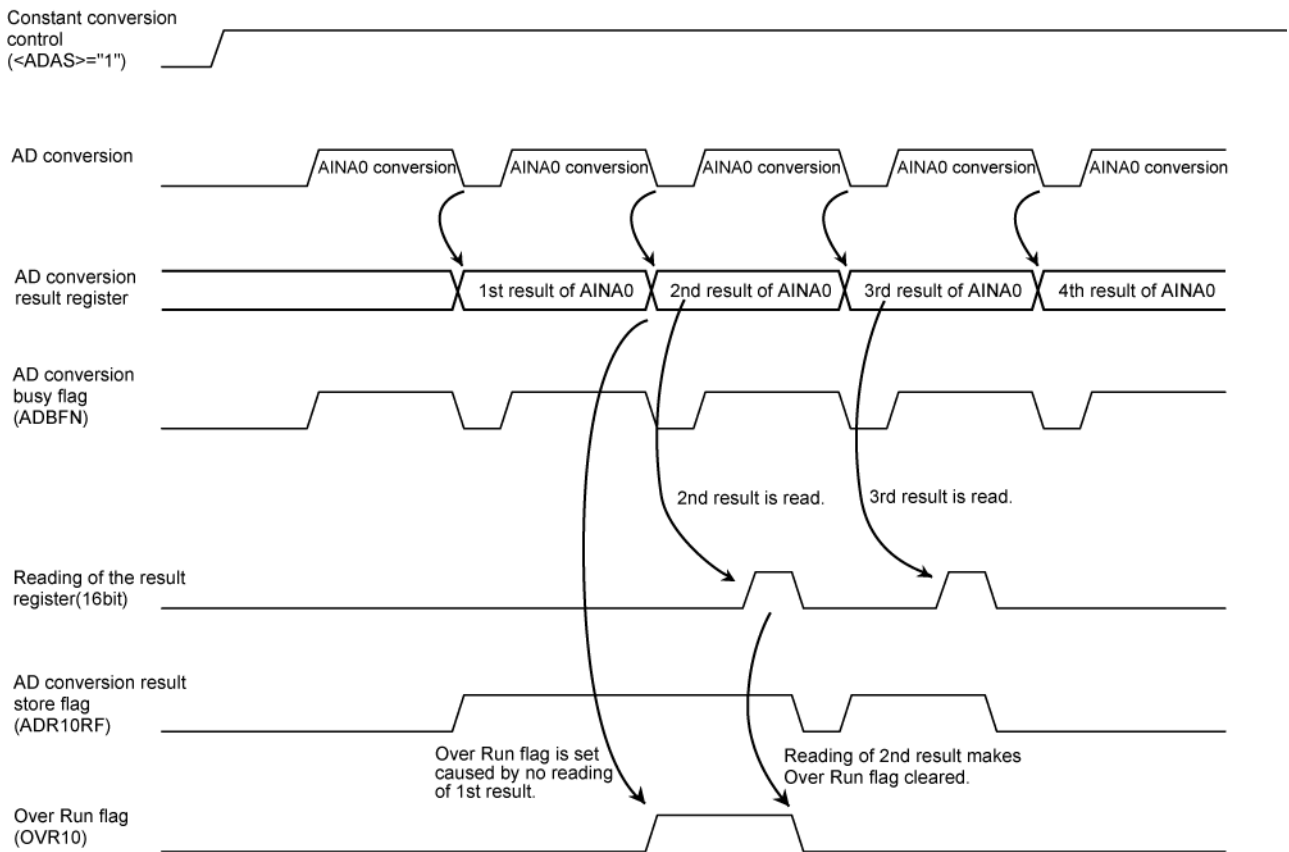


Fig 17-5 Constant conversion



### 17.5.3 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)

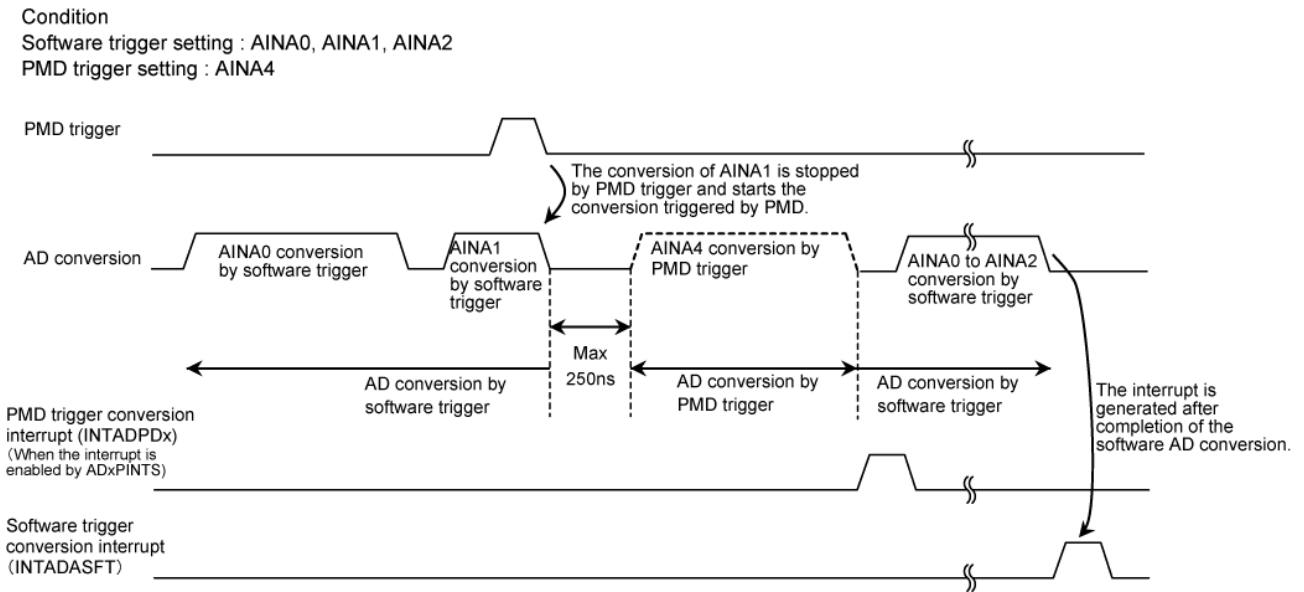


Fig 17-6 AD conversion by PMD trigger

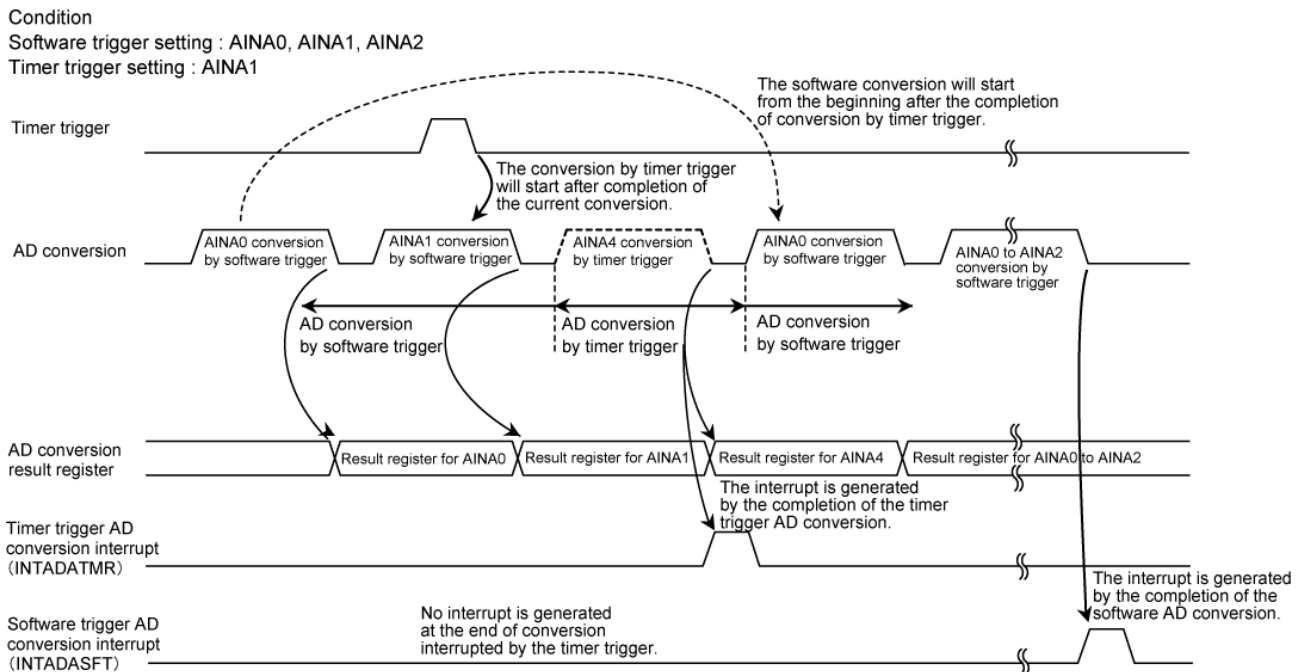


Fig 17-7 AD conversion by timer trigger (1)

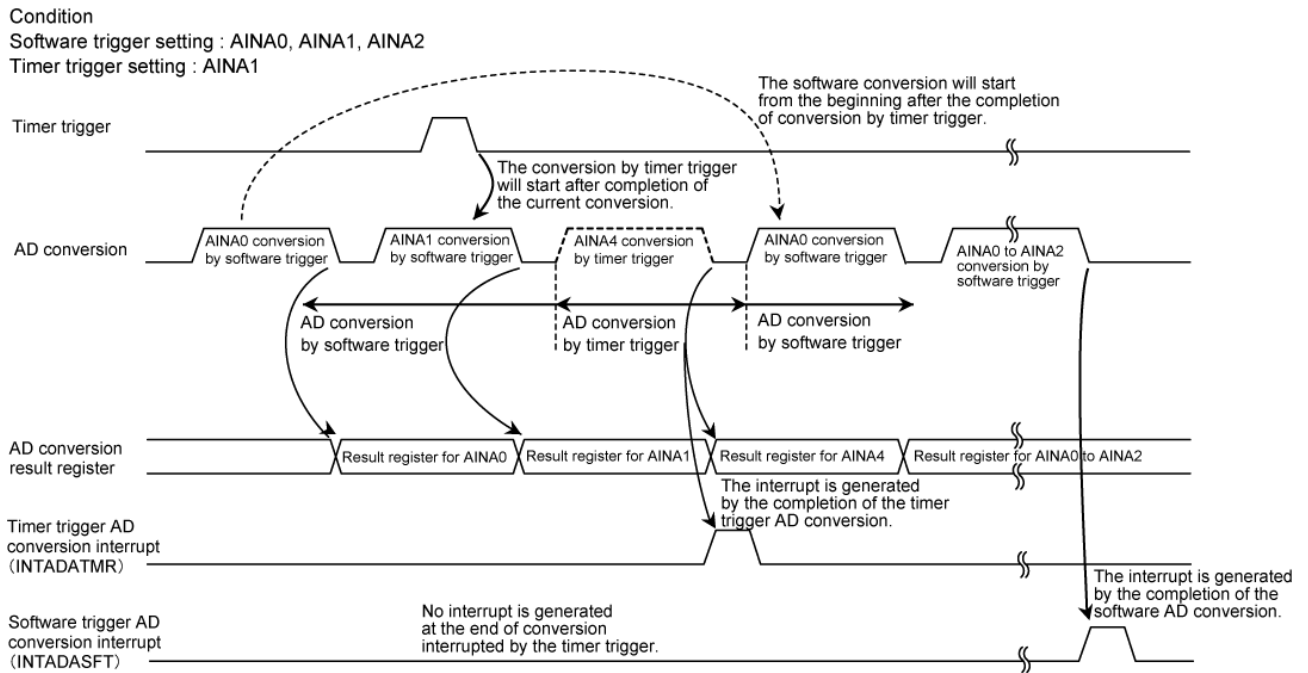
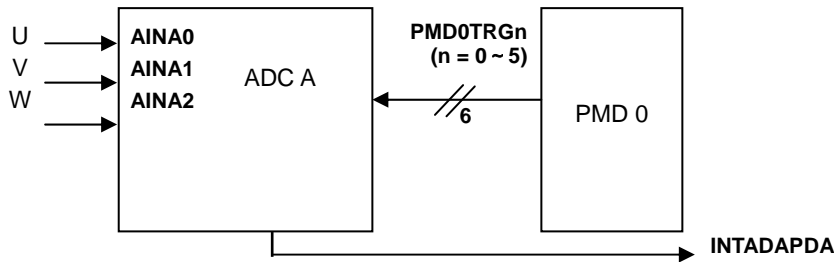


Fig 17-8 AD conversion by timer trigger (2)

## 17.6 Usage Examples

### 17.6.1 Successive Conversion Using PMD A (Three Shunts) and One ADC

The following shows a circuit diagram for AD conversion using one PMD for three shunts and one ADC.



Example ADC settings are shown below.

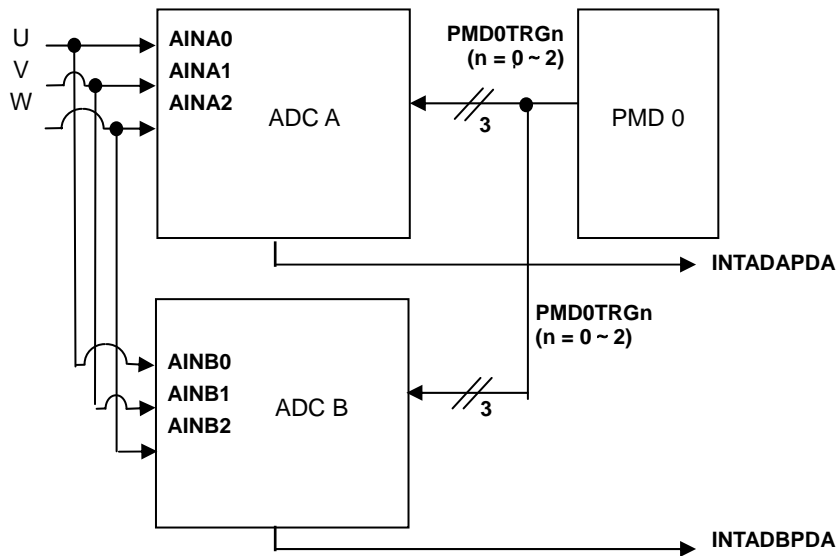
| Program | 0 | 1 | 2 | 3 | 4 | 5 |
|---------|---|---|---|---|---|---|
| reg0    | U | V | W | V | W | U |
| reg1    | V | W | U | U | V | W |
| INT     | A | A | A | A | A | A |

Programs 0 to 5 are assigned to trigger inputs PMD0TRG0 to 5. “reg0” and “reg1” indicate the PMD Trigger Program Registers ADAPSETn0 and ADAPSETn1. “U”, “V” and “W” indicate the phases of a motor. AIN inputs are selected to obtain these phases.

When a trigger input occurs, AD conversion is performed based on reg0 and reg1 sequentially, and then the interrupt signal (INTADAPDA) is generated.

### 17.6.2 Simultaneous Conversion Using One PMD (Three Shunts) and Two ADCs

The following shows a block diagram for AD conversion using one PMD for three shunts and two ADCs.



Example ADC settings are shown below.

ADC A

|         |   |   |   |
|---------|---|---|---|
| Program | 0 | 1 | 2 |
| reg0    | U | V | W |
| INT     | A | A | A |

ADC B

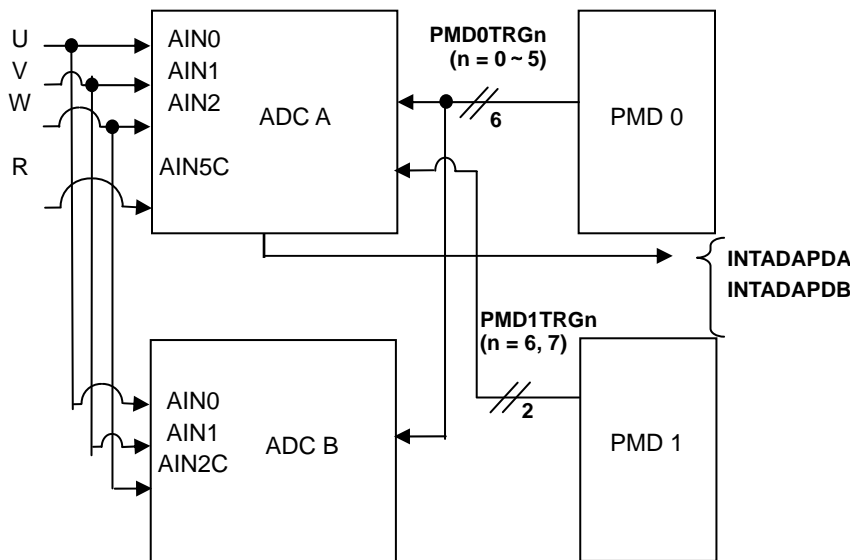
|         |   |   |   |
|---------|---|---|---|
| Program | 0 | 1 | 2 |
| reg0    | V | W | U |
| INT     | A | A | A |

Programs 0 to 2 are assigned to three trigger inputs to ADC A and ADC B. “reg0” indicates the PMD Trigger Program Register ADAPSETn0 and ADBPSETn0. “U”, “V” and “W” indicate the phases of a motor. AIN inputs are selected to obtain these phases.

When a trigger input occurs, ADC A and ADC B are started simultaneously to perform AD conversion based on reg0, and the interrupt signals (INTADAPDA , INTADBPDA) are output to ADC A and ADC B.

### 17.6.3 Successive Conversion Using PMD A (Three Shunts), PMD B (One Shunt) and Two ADCs

The following shows a circuit diagram for AD conversion using one PMD for three shunts, one PMD for one shunt and two ADCs.



Example ADC settings are shown below.

ADC A

| Trigger | PMD0   | PMD0   | PMD0   | PMD1 | PMD1 |
|---------|--------|--------|--------|------|------|
|         | 0,3    | 1,4    | 2,5    | 6    | 7    |
| Program | 0      | 1      | 2      | 3    | 4    |
| reg0    | PMD0 U | PMD0 V | PMD0 W |      |      |
| reg1    |        |        |        | R    |      |
| reg2    |        |        |        |      | R    |
| INT     | A      | A      | A      | -    | B    |

ADC B

| Trigger | PMD0   | PMD0   | PMD0   |
|---------|--------|--------|--------|
|         | 0,3    | 1,4    | 2,5    |
| Program | 0      | 1      | 2      |
| reg0    | PMD0 V | PMD0 W | PMD0 U |
| INT     | -      | -      | -      |

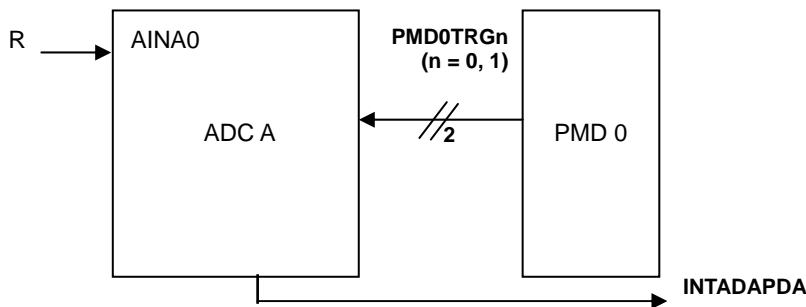
In ADC A, programs 0 to 2 are assigned to six trigger signals from PMD0 and programs 3 and 4 are assigned to two trigger signals from PMD1. In ADC B, programs 0 to 2 are assigned to six trigger signals from PMD0. “reg0”, “reg1” and “reg2” indicate the PMD Trigger Program Registers ADxPSETn0, ADxPSETn1 and ADxPSETn2 (x=A,B : ADC Unit). “U”, “V” and “W” indicate the

phases of a motor. AIN inputs are selected to obtain these phases. “R” indicates a resistor, where the AIN that is connected to that resistor is set.

When a trigger input occurs, ADC A or ADC B is started to perform AD conversion. In ADC A, the interrupt (INTADAPDA) is generated for a trigger from PMD0 and the interrupt (INTADAPDB) is generated for a trigger from PMD1. In ADC B, interrupt generation is disabled in this example.

### 17.6.4 Successive Conversion Using One PMD (One Shunt) and One ADC

The following shows a circuit diagram for AD conversion using one PMD for one shunt and one ADC.



Example ADC settings are shown below.

| Trigger | PMD0 | PMD0 |
|---------|------|------|
|         |      | 0    |
| Program | 0    | 1    |
| reg0    | R    | -    |
| reg1    | -    | R    |
| INT     | -    | A    |

Programs 0 and 1 are assigned to two trigger signals from PMD0. “reg0” and “reg1” indicate the PMD Trigger Program Registers ADAPSETn0 and ADAPSETn1. “R” indicates a resistor, where the AIN input that is connected to that resistor is set.

When a trigger input occurs, the ADC is started to execute programs 0 and 1 sequentially. When program 1 is completed, the interrupt (INTADAPDA) is generated.

## 18 . Op-Amps/Analog Comparators

The TMPM370 has four op-amps and analog comparators. Each op-amp amplifies a voltage received via an input port and feeds its output voltage into a 12-bit successive-approximation analog-to-digital (A/D) converter(s). These op-amps are used to amplify voltage differentials across shunt resistors for motor current measurement. The output of each op-amp is also fed into an analog comparator and compared to the corresponding reference voltage derived from an external resistor. The comparator provides an abnormal current indication to the EMG logic.

Figure 18-1 shows the block diagram of the op-amps/analog converters.

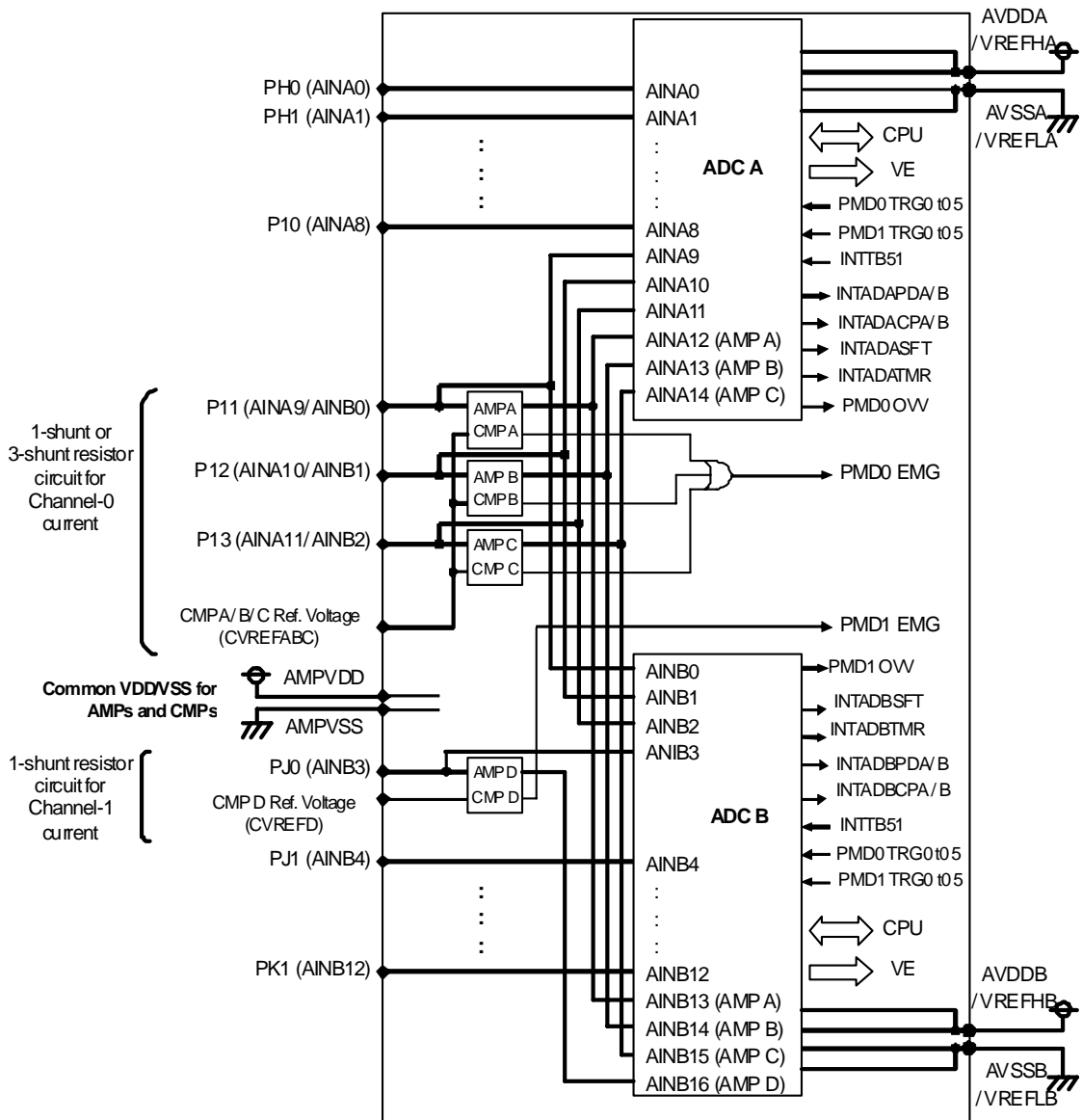


Figure 18-1 Op-Amps/Analog Converters Block Diagram

Op-amps A, B and C (AMP A/B/C) are intended to be used for 3-shunt current sensing. The amplified voltages from AMP A/B/C are fed into two A/D converters to allow simultaneous conversions of two shunt voltages out of three corresponding to the U, V and W phases of a motor.

The inputs of AMP A/B/C are also connected with the A/D converters (AINA 9/10/11, AINB13/14/15) directly; thus, even if AMP A/B/C are disabled, two shunt voltages can be converted into digital values at a time.

Op-amp D (AMP D) only supports 1-shunt current sensing. The amplified voltage from AMP D is fed into one A/D converter (AINB16).

See the block diagram of the op-amps/analog converters shown in Figure 18-1.

Analog comparators A/B/C/D are all connected to op-amps A/B/C/D; thus the comparators can compare an amplified voltage against the reference voltage. Each op-amp can be individually disabled by software; if disabled, the corresponding comparator takes as input a shunt voltage from an input port.

Analog comparators A/B/C, designed to be connected to a 3-shunt resistor circuit, have a common reference voltage (CMPREFABC).

Analog comparator D is designed to be connected to a 1-shunt resistor circuit; it has an independent reference voltage (CMPREFD).

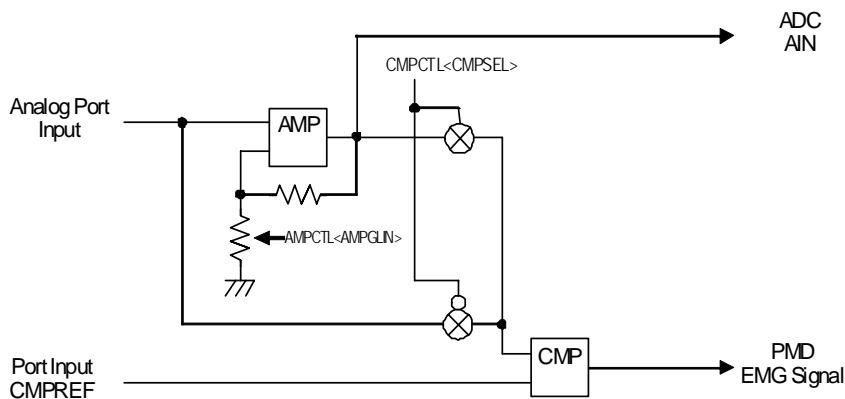


Figure 08-2 Op-Amp/Analog Comparator



## 18.1 Register List

The following table lists the control registers and their addresses for the op-amps and analog comparators.

| Register                      |         | Address     |
|-------------------------------|---------|-------------|
| Amp A Control Register        | AMPCTLA | 0x4003_0400 |
| Amp B Control Register        | AMPCTLB | 0x4003_0408 |
| Amp C Control Register        | AMPCTLC | 0x4003_0410 |
| Amp D Control Register        | AMPCTLD | 0x4003_0418 |
| Comparator A Control Register | CMPCTLA | 0x4003_0420 |
| Comparator B Control Register | CMPCTLB | 0x4003_0428 |
| Comparator C Control Register | CMPCTLC | 0x4003_0430 |
| Comparator D Control Register | CMPCTLD | 0x4003_0438 |

## 18.2 Register Description

The op-amps are individually programmable through the Amp A/B/C/D Control Registers (AMPCTLA/AMPCTLB/AMPCTLC/AMPCTLD), which allow software to enable and disable each op-amp and select a voltage gain from eight levels.

The comparators are individually programmable through the Comparator A/B/C/D Control Registers (CMPCTLA/CMPCTLB/CMPCTLC/CMPCTLD), which allow software to enable and disable each comparator and select its input source (a port input or an op-amp output).

If an external op-amp is used instead of an on-chip op-amp, the on-chip op-amp should be disabled ( $AMPENx = 0$ ) and the comparator should be configured to accept the input port voltage, bypassing the associated op-amp ( $CMPSELx = 0$ ).

The following describes each control register.

Amp A/B/C/D Control Registers

AMPCTLx  
(x: A,B,C,D)

|             | 7                             | 6                             | 5                             | 4                             | 3  | 2        | 1        | 0   |
|-------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--|----------|----------|---|
| Bit Symbol  | /                             |                               |                               |                               | AMPGLIN2   | AMPGLIN1 | AMPGLIN0 | AMPEN                                       |
| Read/Write  | R                             | R                             | R                             | R                             | R/W  |          |          | R/W   |
| Default     | 0                             | 0                             | 0                             | 0                             | 000  |          |          | 0   |
| Description | Reading this bit returns a 0. | Reading this bit returns a 0. | Reading this bit returns a 0. | Reading this bit returns a 0. | Gain Select<br>000: 1.5x 100: 4.0x<br>001: 2.5x 101: 6.0x<br>010: 3.0x 110: 8.0x<br>011: 3.5x 111: 10.0x |          |          | AMP Enable<br>0: Standby<br>1: Enables AMP. |

Comparator A/B/C/D Control Registers

CMPCTLx  
(x: A,B,C,D)

|             | 7                             | 6                             | 5                             | 4                             | 3                             | 2                             | 1   | 0   |
|-------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|---|---|
| Bit Symbol  | /                             |                               |                               |                               |                               |                               | CMPSEL  | CMPEN                                       |
| Read/Write  | R                             | R                             | R                             | R                             | R                             | R                             | R/W   | R/W   |
| Default     | 0                             | 0                             | 0                             | 0                             | 0                             | 0                             | 0   | 0   |
| Description | Reading this bit returns a 0. | Reading this bit returns a 0. | Reading this bit returns a 0. | Reading this bit returns a 0. | Reading this bit returns a 0. | Reading this bit returns a 0. | Input Select<br>0: Input port<br>1: Op-amp output | CMP Enable<br>0: Standby<br>1: Enables CMP. |

## 19. Encoder Input Circuit (ENC)

The TMPM370FY has a two-channel incremental encoder interface (ENC0,1), which can determine the direction and the absolute position of a motor, based on input signals from an incremental encoder.

The discussions in this chapter apply to ENC0. For ENC1, the names of registers, interrupt signals and pins in the following text should be replaced as appropriate, as shown in Table 19.1 to Table 19.3.

Table 19.1 List of the ENC Registers

| Register                        | ENC0            |             | ENC1            |             |
|---------------------------------|-----------------|-------------|-----------------|-------------|
|                                 | Register Symbol | Address     | Register Symbol | Address     |
| Encoder Input Control Register  | EN0TNCR         | 0x4001_0400 | EN1TNCR         | 0x4001_0500 |
| Encoder Counter Reload Register | EN0RELOAD       | 0x4001_0404 | EN1RELOAD       | 0x4001_0504 |
| Encoder Compare Register        | EN0INT          | 0x4001_0408 | EN1INT          | 0x4001_0508 |
| Encoder Counter                 | EN0CNT          | 0x4001_040C | EN1CNT          | 0x4001_050C |

Table 19.2 Interrupt Sources

| Interrupt Source | ENC0    | ENC1    |
|------------------|---------|---------|
| ENC interrupt    | INTENC0 | INTENC1 |

Table 19.3 Pin Names

| Pin                 | ENC0      | ENC1      |
|---------------------|-----------|-----------|
| Channel A input pin | PD0/ENCA0 | PF2/ENCA1 |
| Channel B input pin | PD1/ENCB0 | PF3/ENCB1 |
| Channel Z input pin | PD2/ENCZ0 | 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

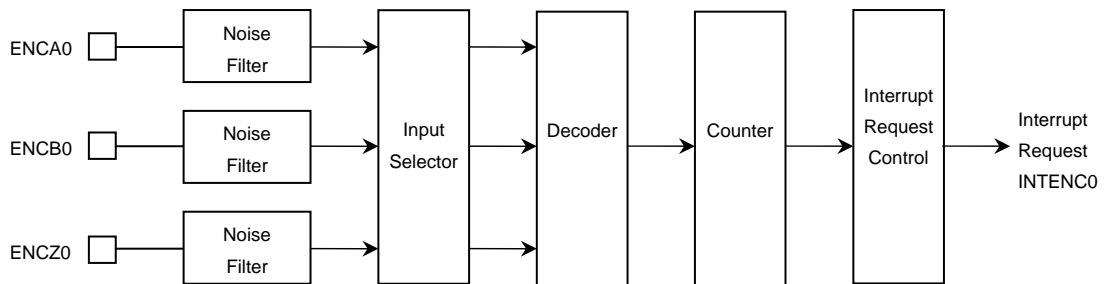


Figure 19.1 ENC Block Diagram

### 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 0 Input Control Register

ENOTNCR  
(0x4001\_0400)

|             | 31  | 30   | 29  | 28  | 27   | 26   | 25  | 24   |  |
|-------------|---|--|---|---|--|--|---|--|--|
| 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<br>00: Encoder mode<br>01: Sensor Event Count mode<br>10: Sensor Timer Count mode<br>11: Timer mode             |   | [In Sensor mode]<br>2/3-Phase Input Select<br>0: 2-phase<br>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<br>0: -<br>1: Counter compared<br><br>This bit is cleared on a read. | [In Sensor Timer Count mode]<br>Revolution Error<br>0: -<br>1: Error occurred.<br><br>This bit is cleared on a read. | Rotation Direction<br>0: CCW<br>1: CW   | Z_ Detected<br>0: Not detected<br>1: Z phase detected | [In Sensor Timer Count and Timer modes]<br>Software Capture<br>0: -<br>1: Software capture | Encoder Counter Clear<br>0: -<br>1: Clears the counter.  | [In Timer mode]<br>Z Trigger Edge Select<br>0: Rising edge<br>1: Falling edge | Compare Enable<br>0: Compare disabled<br>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<br>0: Disabled<br>1: Enabled                                       | ENC Run<br>0: Disabled<br>1: Enabled   | Noise Filter<br>00: No filtering<br>01: Filters out pulses narrower than 31/fsys (387.5 ns@80 MHz).<br>10: Filters out pulses narrower than 63/fsys(787.5 ns@80 MHz).<br>11: Filters out pulses narrower than 127/fsys(1587 ns@80 MHz). |   | ENC Interrupt Enable<br>0: Disabled<br>1: Enabled  | Encoder Pulse Division Factor<br>000: (1)<br>001: (2)<br>010: (4)<br>011: (8)<br>100: (16)<br>101: (32)<br>110: (64)<br>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 EN0INT 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 ENOCNT 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 ENOINT 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 (ENOCNT) 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 ENOINT 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<br>(After reset) | When <ENRUN> = 1<br>(During active operation)     | When <ENRUN> = 0<br>(During idle mode)           | How to clear a counter or flag when <ENRUN> = 0 |
|--------------------------------|-----------------------------------|---|--|---|
| Encoder counter                | 0x000000                          | Counting  | Keeps the current value.                         | Software clear<br>(Write a 1 to <ENCLR>.)       |
| Noise filter counter           | 0y0000000                         | Counting up                                       | Counting up<br>(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 0 Counter Reload Register

| ENORELOAD<br>(0x4001_0404) |  | 31       | 30       | 29       | 28       | 27       | 26       | 25      | 24      |
|----------------------------|--|----------|----------|----------|----------|----------|----------|---------|---------|
| 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))<br>0x0000 thru. 0xFFFF<br>When Z is used: Specifies the number of counter pulses per revolution.<br>When Z is not used: Specifies the number of counter pulses per revolution minus 1. |          |          |          |          |          |          |         |         |

Description:

<RELOAD15:0>: setting the Encoder Counter Period

- 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 ENORELOAD register is only used in Encoder mode.

Encoder 0 Compare Register

|                         |  |       |       |       |       |       |       |       |
|-------------------------|--|-------|-------|-------|-------|-------|-------|-------|
| EN0INT<br>(0x4001_0408) | 31   | 30    | 29    | 28    | 27    | 26    | 25    | 24    |
| 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.<br><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:<br>Generates an interrupt at the programmed encoder pulse count (0x0000 thru. 0xFFFF).<br>In Sensor Event Count mode:<br>Generates an interrupt at the programmed encoder pulse count (0x0000 thru. 0xFFFF).<br>In Sensor Timer Count mode:<br>Generates an interrupt when the counter has reached the programmed value without detecting a pulse (0x000000 thru. 0xFFFFF).<br>In Timer mode:<br>Generates an interrupt when the counter value has reached the programmed value (0x000000 thru. 0xFFFFF). |       |       |       |       |       |       |       |

Description:

<INT15:0>: Counter Compare Value

- 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 0 Counter Register

|                         |   |       |       |       |       |       |       |       |       |
|-------------------------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| EN0CNT<br>(0x4001_040C) |   | 31    | 30    | 29    | 28    | 27    | 26    | 25    | 24    |
| 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.<br><CNT23:16> are used only in Sensor Timer Count mode and Timer mode.<br>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:<br>Number of encoder pulses<br>0x0000 thru. 0xFFFF<br>In Sensor mode:<br>Pulse detection time or the encoder counter value captured under software control<br>0x000000 thru. 0xFFFFFFFF<br>In Timer mode:<br>encoder counter value captured by hardware signaling or under software control<br>0x000000 thru. 0xFFFFFFFF |       |       |       |       |       |       |       |       |

Description:

<CNT15:0>: Encoder Counter/Captured Value

- 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 reached 0xFFFF, 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.

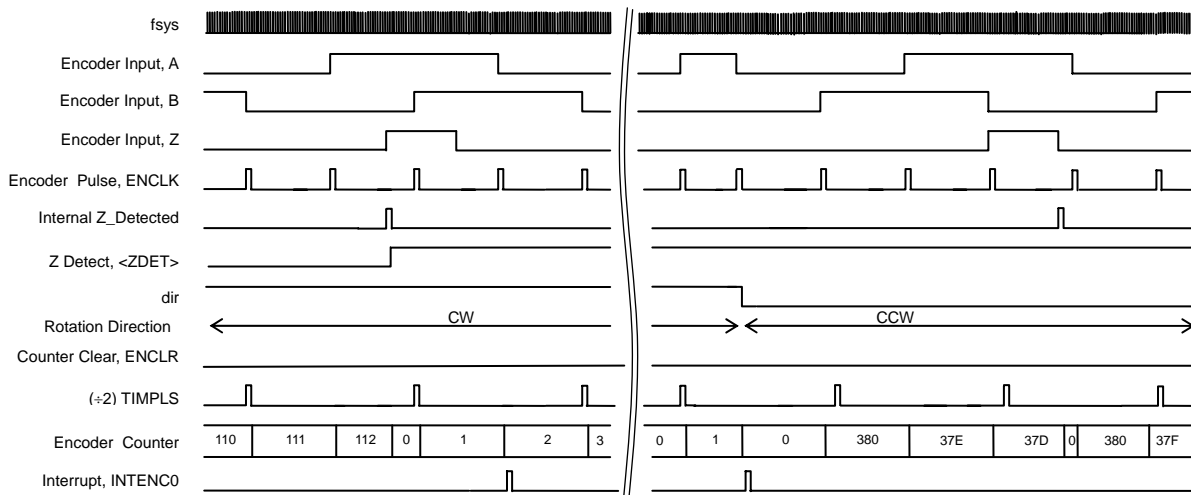
- In Sensor Timer Count mode
  - <CNT23:0> contains the value of the encoder counter captured by either the encoder pulse (ENCLK) or software setting <SFTCAP> to 1. The captured value in <CNT23:0> is cleared to 0 on system reset. It can also be cleared by clearing the counter via setting <ENCLR> to 1 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 0xFFFFFFFF, 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 0xFFFFFFFF, it wraps around to 0 automatically.

## 19.3 Functional Description

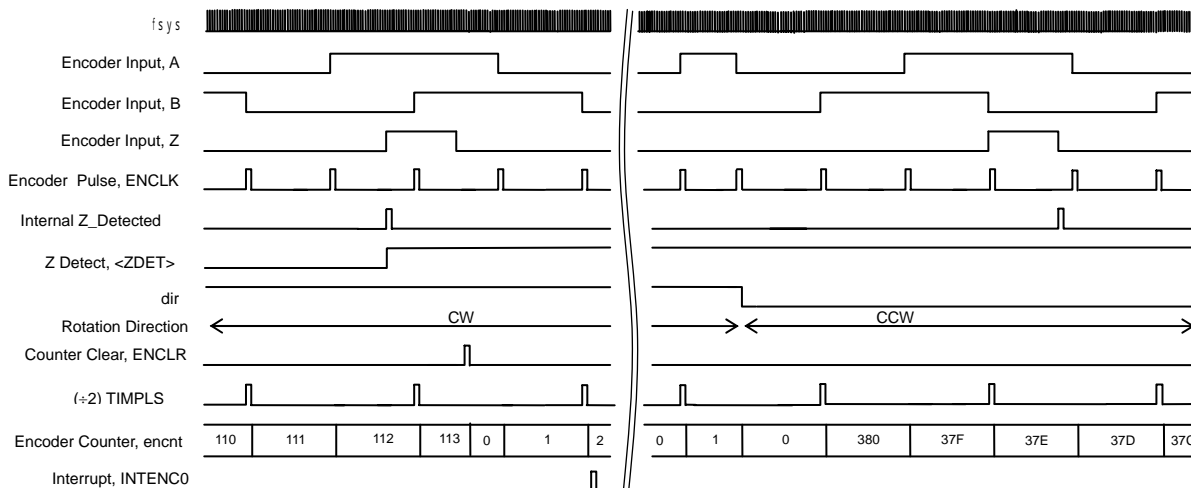
### 19.3.1 Operating Modes

#### 19.3.1.1 Encoder Mode

(1) When  $\langle ZEN \rangle = 1$  ( $\langle RELOAD \rangle = 0x0380$ ,  $\langle EN0INT \rangle = 0x0002$ )



(2) When  $\langle ZEN \rangle = 0$  ( $\langle RELOAD \rangle = 0x0380$ ,  $\langle EN0INT \rangle = 0x0002$ )

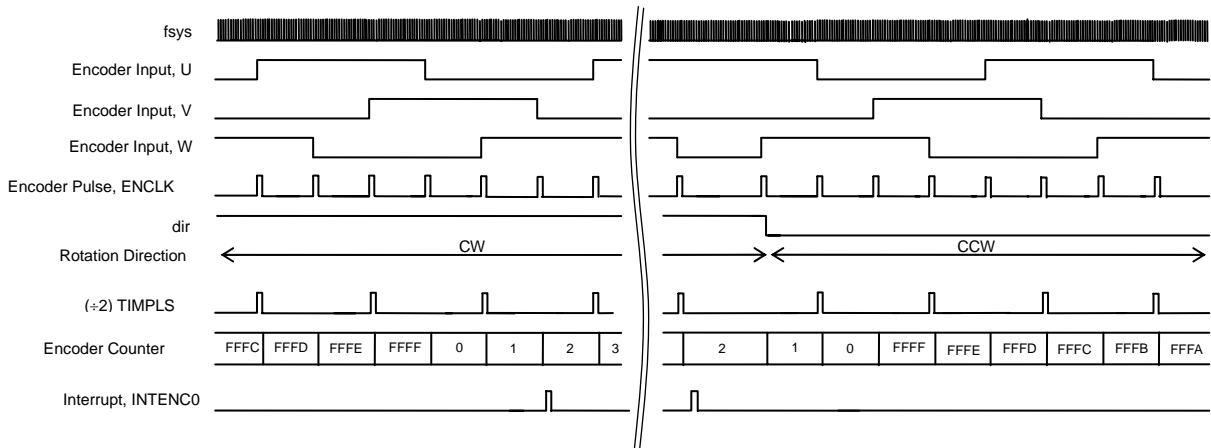


- In Encoder mode, the incremental encoder inputs of the TMPM370 should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied\_by\_4 clock derived from the decoded A and B quadrature signals.
- During CW rotation (i.e., when A leads B), the encoder counter counts up; when it has reached the value of  $\langle RELOAD \rangle$ , 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  $\langle RELOAD \rangle$  on the next ENCLK.

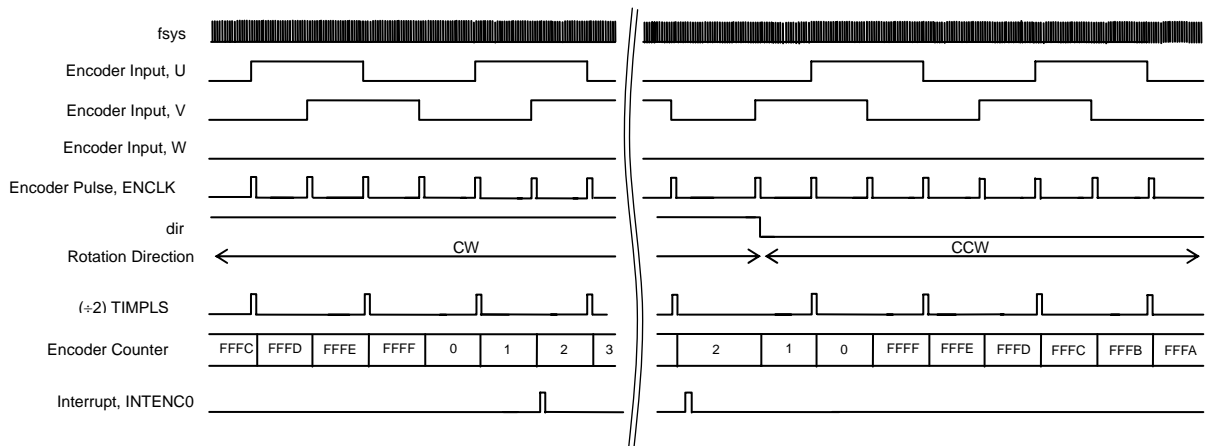
- Additionally, when  $\langle ZEN \rangle = 1$ , the encoder counter is cleared to 0 on the rising edge of Z during CW rotation and on the falling edge of Z during CCW rotation (at the internal Z\_Detected timing). If the ENCLK and Z edges coincide, the encoder counter is cleared to 0 without incrementing or decrementing.
- When  $\langle ENCLR \rangle$  is set to 1, causing the encoder counter to be cleared to 0.
- $\langle UD \rangle$  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  $\langle CMPEN \rangle$  is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of  $\langle EN0INT \rangle$ . When  $\langle ZEN \rangle = 1$ , however, an interrupt does not occur while  $\langle ZDET \rangle = 0$ .
- Clearing  $\langle ENRUN \rangle$  to 0 clears  $\langle ZDET \rangle$  and  $\langle UD \rangle$  to 0.

### 19.3.1.2 Sensor Event Count Mode

(1) When  $\langle P3EN \rangle = 1$  ( $\langle EN0INT \rangle = 0x0002$ )



(2) When  $\langle P3EN \rangle = 0$  ( $\langle EN0INT \rangle = 0x0002$ )



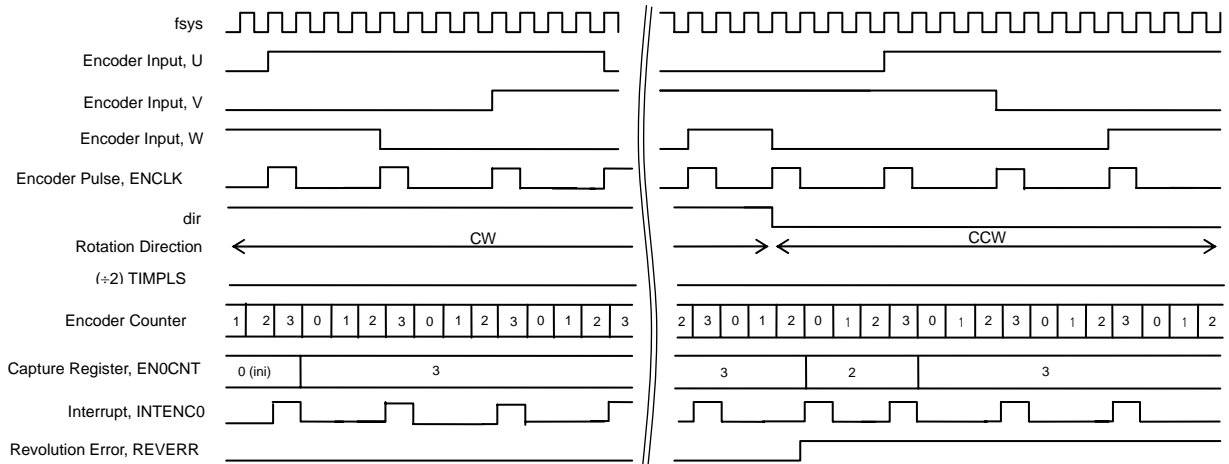
- In Sensor Event Count mode, the Hall sensor inputs of the TMPM370 should be connected to the U, V and W channels. The encoder counter counts the pulses of ENCLK, which is either multiplied\_by\_4 clock (when  $\langle P3EN \rangle = 0$ ) derived from the decoded U and V signals or multiplied\_by\_6 clock (when  $\langle P3EN \rangle = 1$ ) 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



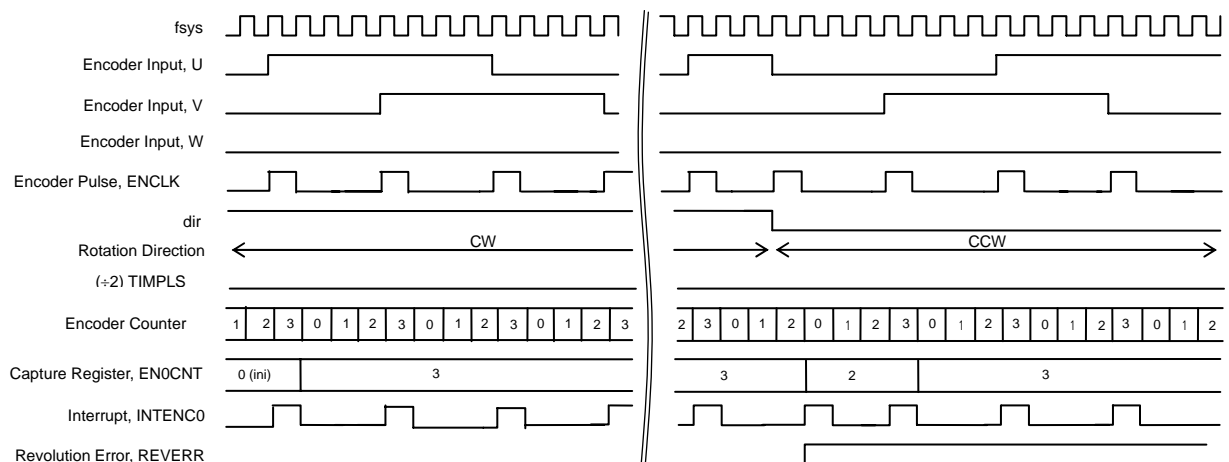
- to 0 on the next ENCLK.
- During CCW rotation, the encoder counter counts down; when it has reached 0x0000, it wraps around to 0xFFFF on the next ENCLK.
- When <ENCLR> is set to 1, causing the internal counter to be cleared to 0.
  - <UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- 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 <EN0INT>.
- Clearing <ENRUN> to 0 clears <UD> to 0.

**19.3.1.3 Sensor Timer Count Mode**

(1) When <P3EN> = 1 (<EN0INT> = 0x0002)



(2) When <P3EN> = 0 (<EN0INT> = 0x0002)

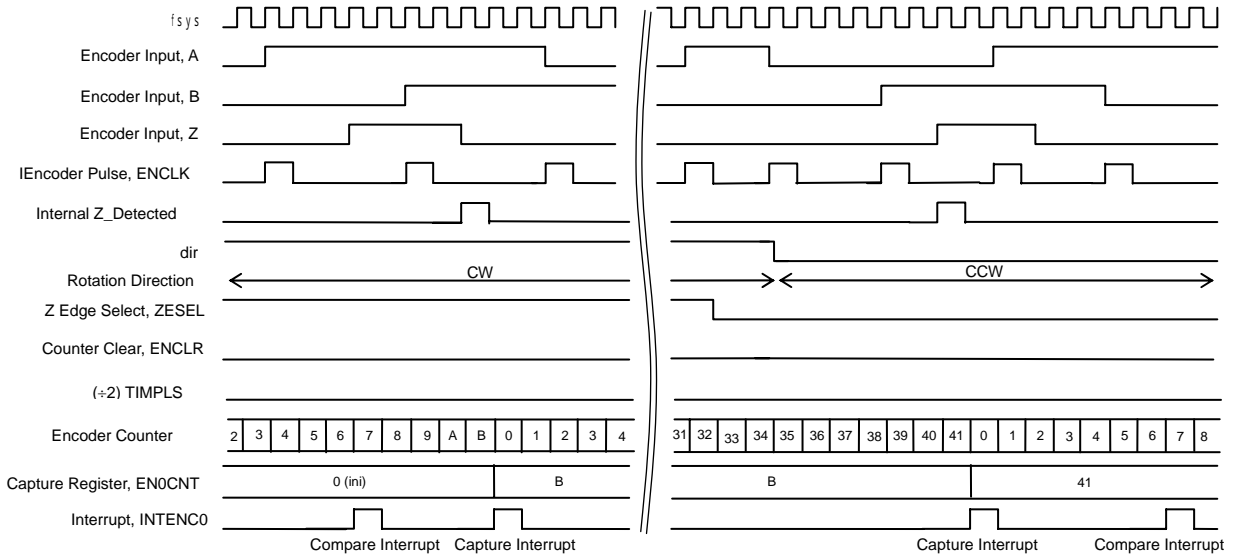


- In Sensor Timer Count mode, the Hall sensor inputs of the TMPM370 should be connected to the U, V and W channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied\_by\_4 clock (when <P3EN> = 0) derived from the decoded U and V signals or multiplied\_by\_6 clock (when <P3EN> = 1) derived from the decoded U, V and W signals.

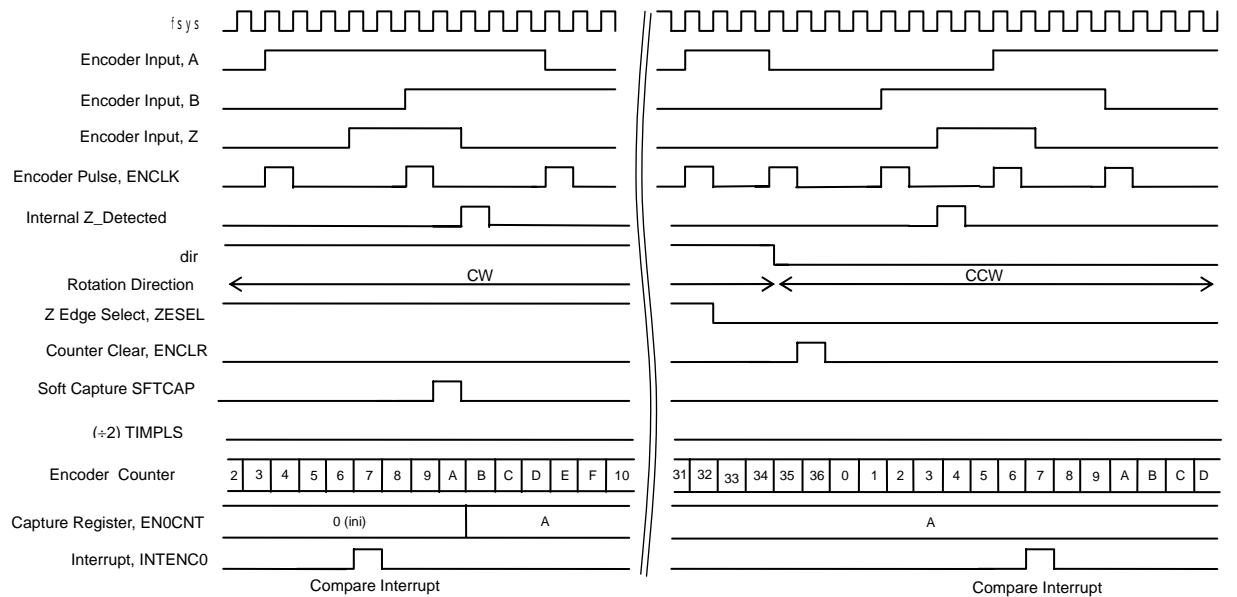
- The encoder counter always counts up; it is cleared to 0 on ENCLK. When the encoder counter has reached 0xFFFFFFFF, it wraps around to 0.
- When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.
- ENCLK causes the value of the encoder counter to be captured into the EN0CNT register. The captured counter value can be read out of EN0CNT.
- 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 <EN0INT>.
- <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 (<EN0INT> = 0x0006)



(2) When <ZEN> = 0 (<EN0INT> = 0x0006)



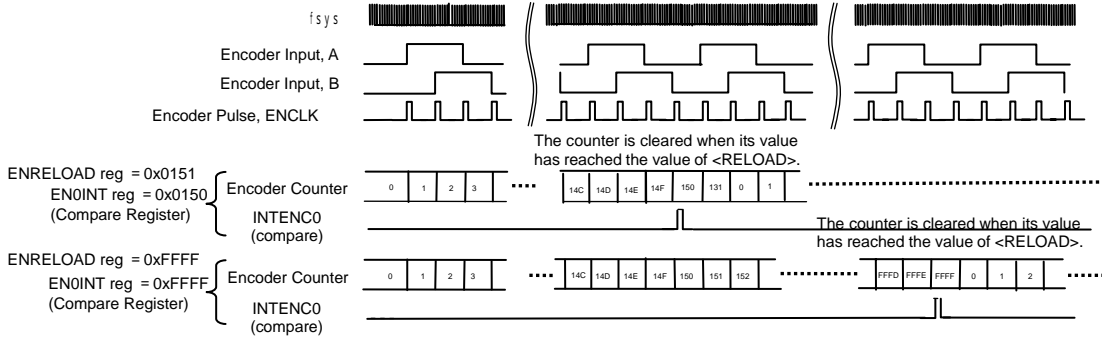
- When <ZEN> = 1, the Z input pin is used as an external trigger. When <ZEN> = 0, no external input is used to trigger the timer.
- The encoder counter always counts up. If <ZEN> = 1, the counter is cleared to 0 on the selected edge of Z (at the internal Z\_Detected timing): a rising edge when <ZESEL> = 0 and a falling edge when <ZESEL> = 1. When the encoder counter has reached 0xFFFFF, 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.

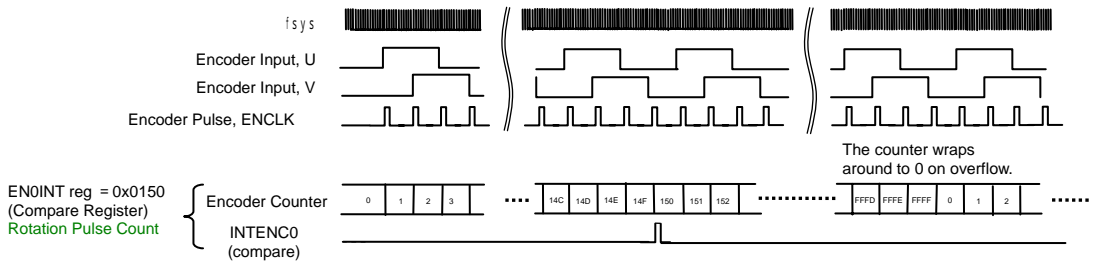
- <UD> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <ENINT>.
- Clearing <ENRUN> to 0 clears <UD> to 0.
- The value of the ENCNT register (the captured value) is retained, 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

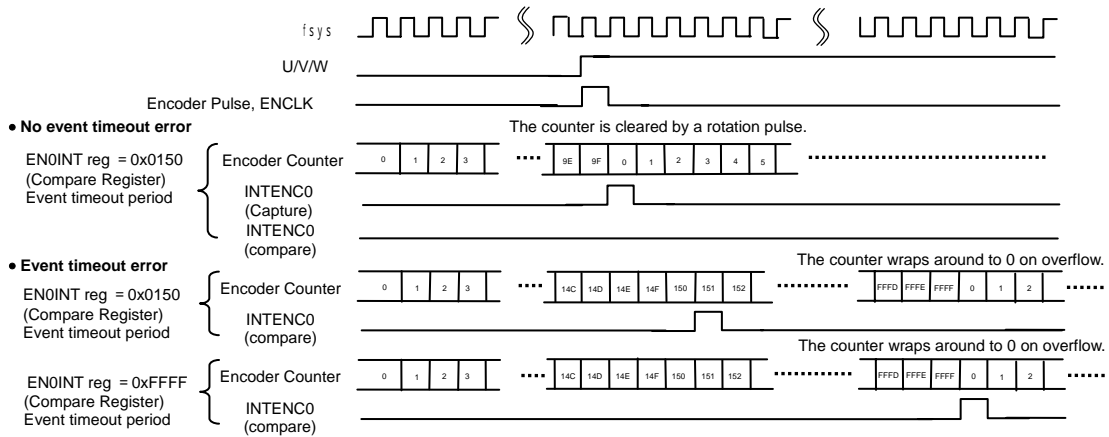
#### 19.3.2.1 Encoder Mode



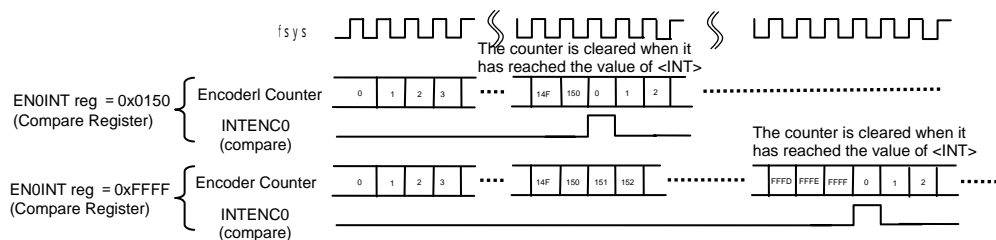
#### 19.3.2.2 Sensor Event Count Mode



#### 19.3.2.3 Sensor Timer Count Mode

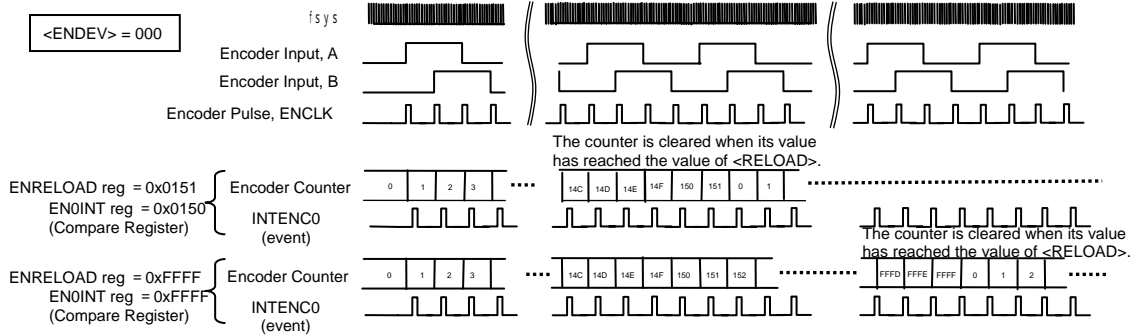


#### 19.3.2.4 Timer Mode

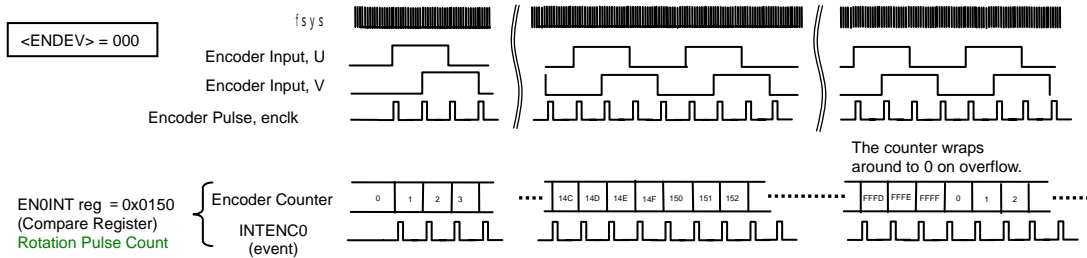


### 19.3.3 Counter Operation and Interrupt Generation When <CMPEN> = 0

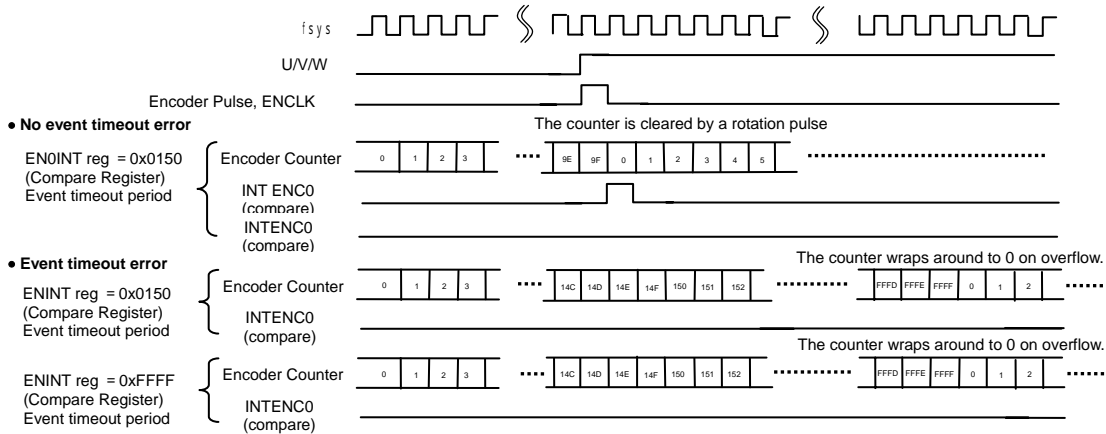
#### 19.3.3.1 Encoder Mode



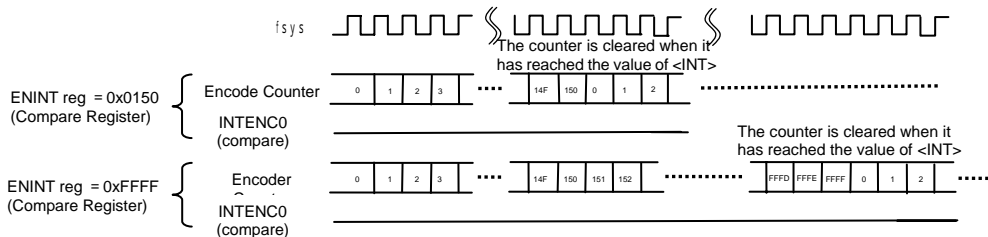
#### 19.3.3.2 Sensor Event Count Mode



#### 19.3.3.3 Sensor Timer Count Mode



#### 19.3.3.4 Timer Mode



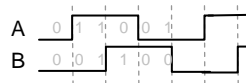
### 19.3.4 Encoder Rotation Direction

The following diagrams illustrate the phase shifting of the A, B and Z channels.

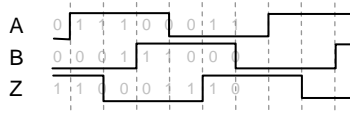
The ENC can interface with both two-phase (A/B) and three-phase (A/B/Z) encoder inputs. For three-phase encoder inputs, <P3EN> should be set to 1.

- (1) Here are possible combinations of values of the A (U), B (V) and Z (W) signals during CW rotation.

- For two-phase inputs

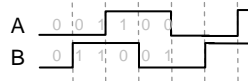


- For three-phase inputs

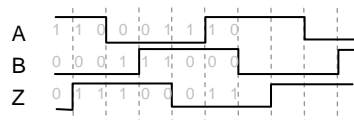


- (2) Here are possible combinations of values of the A (U), B (V) and Z (W) signals during CCW rotation.

- 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<br><MODE1:0>      | <ENR> | <REN> | Input Pins | Count                 | Up/Down | Counter Clear Conditions  | Counter Reload Conditions   | Counter Range (Reload Value) |
|----------------------------------|-------|-------|------------|-----------------------|---------|---|---|------------------------------|
| Encoder Mode<br>00               | 0     | 0     | A, B       | Encoder pulse (ENCLK) | Up      | [1] <ENCLR> is set to 1.<br>[2] counter=<RELOAD>                                | -   | 0x0000 thru.<br><RELOAD>     |
|                                  |       |       |            |                       | Down    | [1] <ENCLR> is set to 1.  | [1] counter=0x0000  |                              |
|                                  | 1     | 0     | A, B, Z    | Encoder pulse (ENCLK) | Up      | [1] <ENCLR> is set to 1.<br>[2] counter=<RELOAD><br>[3] Z trigger               | -   |                              |
|                                  |       |       |            |                       | Down    | [1] <ENCLR> is set to 1.  | [1] counter=0x0000  |                              |
| Sensor Event Count<br>Mode<br>01 | 0     | 0     | U, V       | Encoder pulse (ENCLK) | Up      | [1] <ENCLR> is set to 1.<br>[2] counter=16'hFFFF                                | -   | 0x0000 thru.<br>0xFFFF       |
|                                  |       |       |            |                       | Down    | [1] <ENCLR> is set to 1.  | [1] counter=0x0000  |                              |
|                                  | 1     | 0     | U, V, W    | Encoder pulse (ENCLK) | Up      | [1] <ENCLR> is set to 1.<br>[2] counter=0xFFFF                                  | -   |                              |
|                                  |       |       |            |                       | Down    | [1] <ENCLR> is set to 1.  | [1] counter=0x0000  |                              |
| Sensor Timer Count<br>Mode<br>10 | 0     | 0     | U, V       | fsys                  | Up      | [1] <ENCLR> is set to 1.<br>[2] counter=0xFFFFFFFF<br>[3] Encoder pulse (ENCLK) | -   | 0x000000 thru.<br>0xFFFFFFFF |
|                                  |       |       | U, V, W    | fsys                  | Up      |   |   |                              |
| Timer Mode<br>11                 | 0     | X     | -          | fsys                  | Up      | [1] <ENCLR> is set to 1.<br>[2] counter=0xFFFFFFFF<br>[3] counter=<EN0INT>      | -   | 0x000000 thru.<br>0xFFFFFFFF |
|                                  |       |       | 1          | Z                     | fsys    | Up  | [1] <ENCLR> is set to 1.<br>[2] counter=0xFFFFFFFF<br>[3] counter=<EN0INT><br>[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 .

Table19.5 Interrupt Sources

|   | Interrupt Source                      | Description  | Operating Mode                            | Interrupt Generation                   | Status Flag |
|---|---------------------------------------|--|---|--|-------------|
| 1 | Event counter interrupt               | When <CMPEN> = 1, the encoder counter counts events (encoder pulses). When it has reached the value programmed in <ENOINT>, an interrupt occurs.   | Encoder mode and Sensor Event Count modes | When <INTEN> = 1<br>and<br><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 <ENOINT>, an interrupt occurs. |   | When <INTEN> = 1<br>and<br><CMPEN> = 1 | <CMP>       |
| 5 | Timer compare interrupt               | When <CMPEN> = 1, an interrupt occurs when the timer has reached the value programmed in <ENOINT>.   | Timer mode                                | When <INTEN> = 1<br>and<br><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 TMPM370FY 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 TMPM370FY 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 |            |            |            |
| TMPM370FY    | 256KB       | 0                   | 3    | 1    | 2    | 64         | 1.28sec    | 0.4sec     |

**(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.

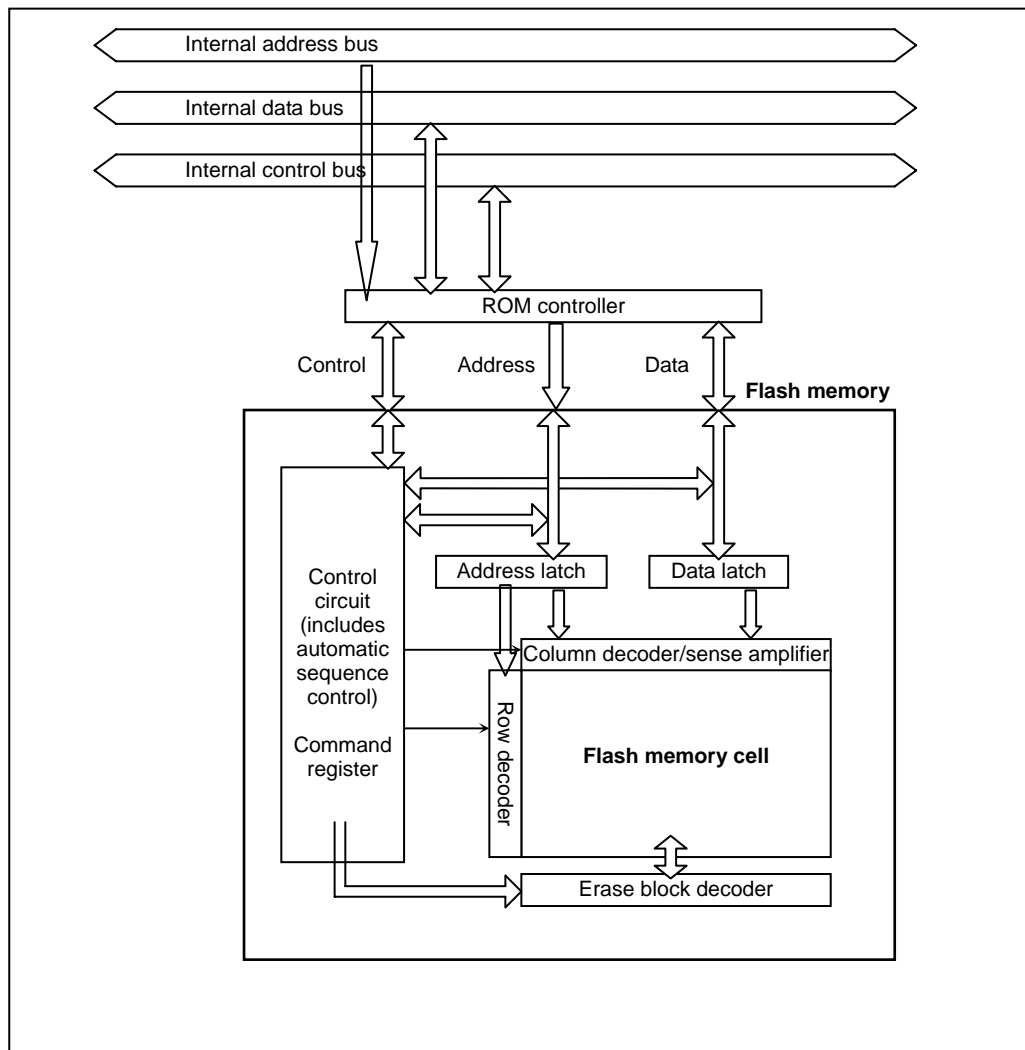
#### Rewriting method

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.

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.

| JEDEC compliant functions  | Modified, added, or deleted functions   |
|--|---|
| <ul style="list-style-type: none"> <li>• Automatic programming</li> <li>• Automatic chip erase</li> <li>• Automatic block erase</li> </ul> | <Modified> Block protect (only software protection is supported)<br><Deleted> Erase resume - suspend function |
| <ul style="list-style-type: none"> <li>• Data polling/toggle bit</li> </ul>  |   |

**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."<br><br>The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0."<br>The user should prepare a routine as part of the application program to make the decision on the selection of the modes. |
| User boot mode   |  |
| 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.

Note: Regardless using or not using Single Boot mode, password must be written in password area.

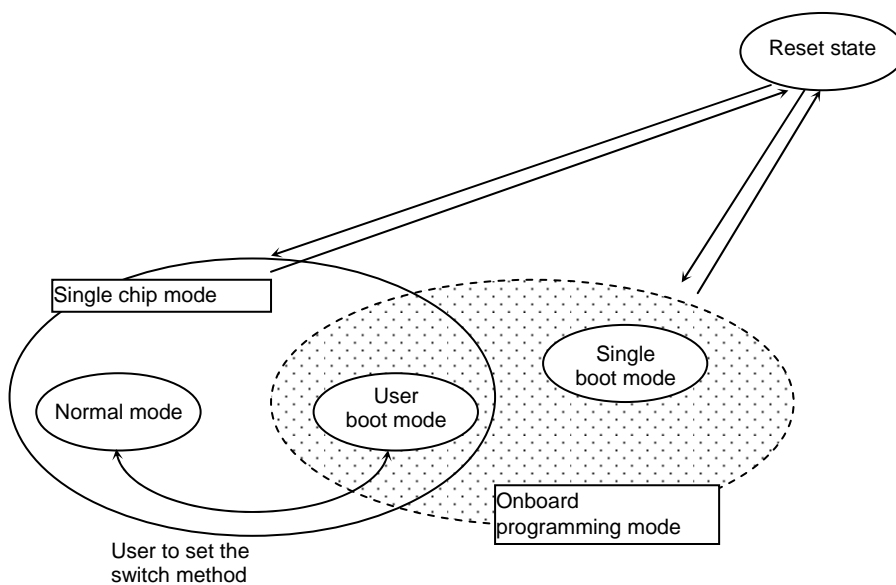
Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the  $\overline{\text{BOOT}}$  (PF0) pin while the device is in reset status.

After the level is set, the CPU starts operation in the selected operation mode when the reset condition is removed. Regarding the  $\overline{\text{BOOT}}$  (PF0) pin, be sure not to change the levels during operation once the mode is selected.

The mode setting method and the mode transition diagram are shown below:

**Table 20-2 Operation Mode Setting**

| Operation mode   | Pin                       |                                |
|------------------|---------------------------|--------------------------------|
|                  | $\overline{\text{RESET}}$ | $\overline{\text{BOOT}}$ (PF0) |
| Single chip mode | 0 → 1                     | 1                              |
| Single boot mode | 0 → 1                     | 0                              |



**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  $\mu\text{s}$  with 80MHz operation; the "1/1" clock gear mode is applied after reset).

**(Note 1)** Regarding to power-on, refer to section "Power On Reset" and "Electric Characteristics".

**(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 TMPM370 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 19.3 On-board Programming of Flash Memory (Rewrite/Erase).

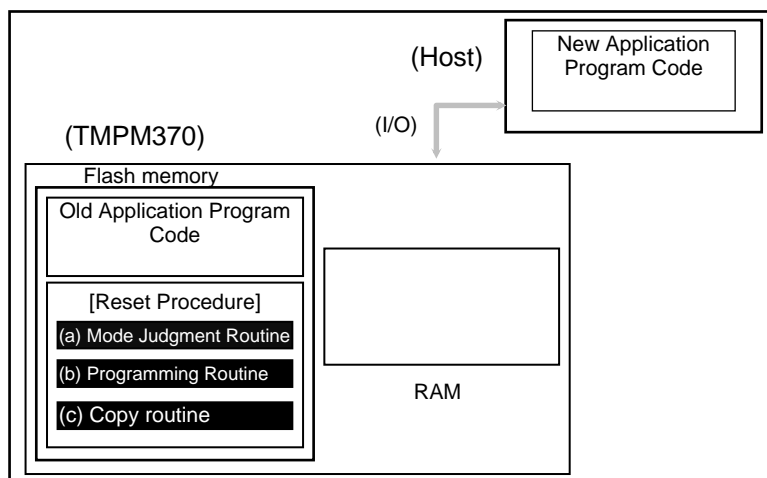
**User Boot Mode**

(1-A) Method 1: Storing a Programming Routine in the Flash Memory

**(Step-1)**

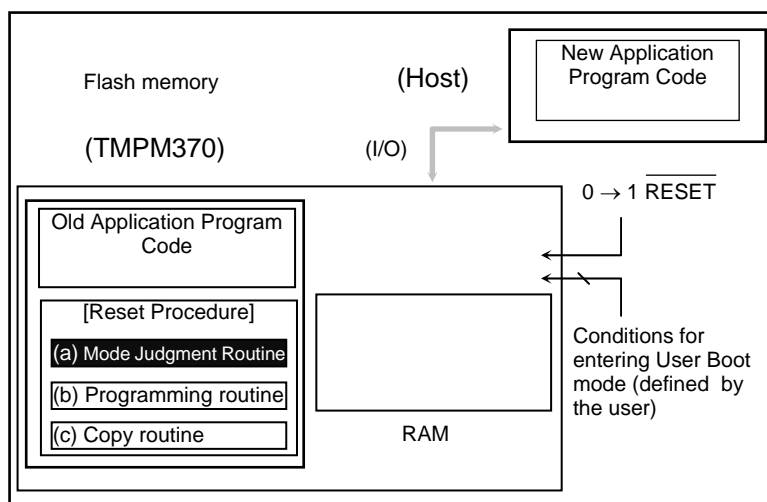
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM370 on a printed circuit board, write the following program routines into an arbitrary 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 TMPM370 flash memory to either the TMPM370 on-chip RAM or external memory device.



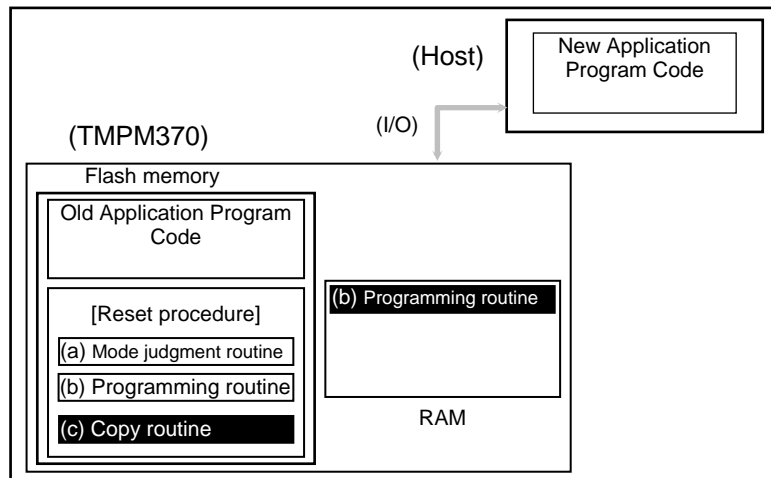
**(Step-2)**

After  $\overline{\text{RESET}}$  is released, the reset procedure determines whether to put the TMPM370 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.)



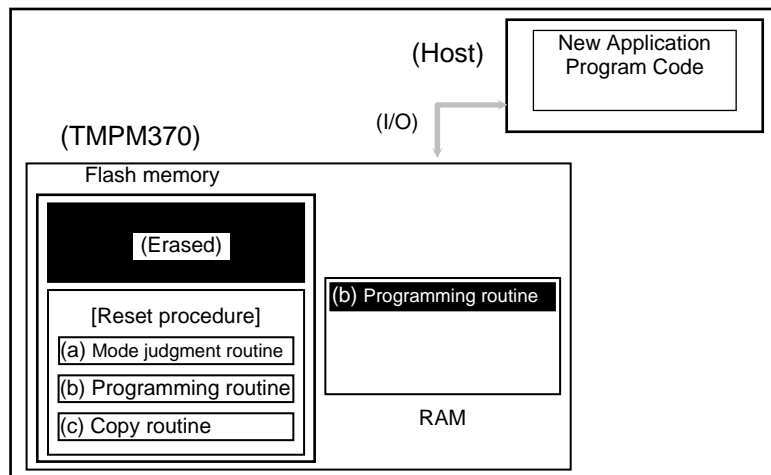
**(Step-3)**

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM370 on-chip RAM.



**(Step-4)**

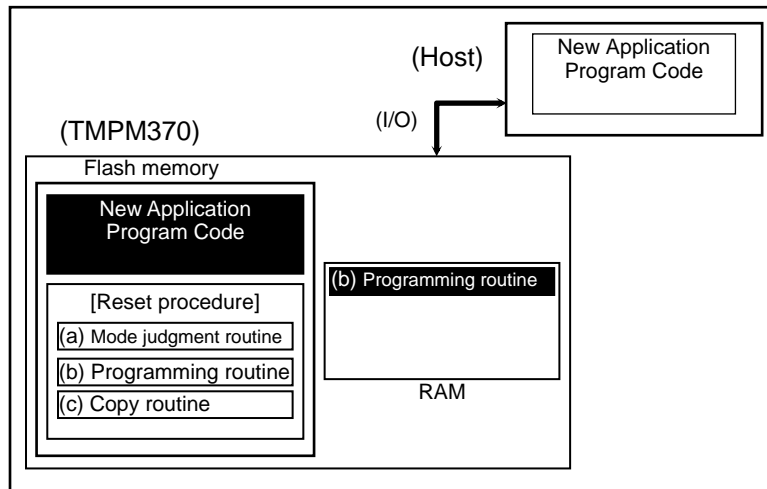
Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.





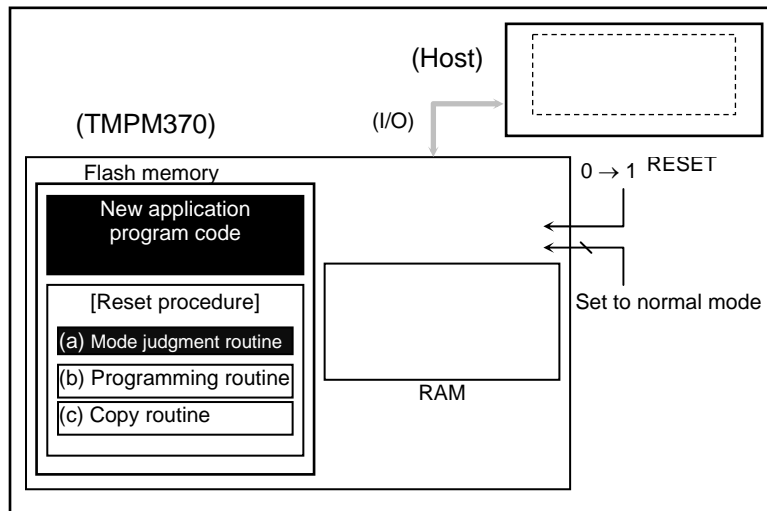
**(Step-5)**

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 TMPM370. 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

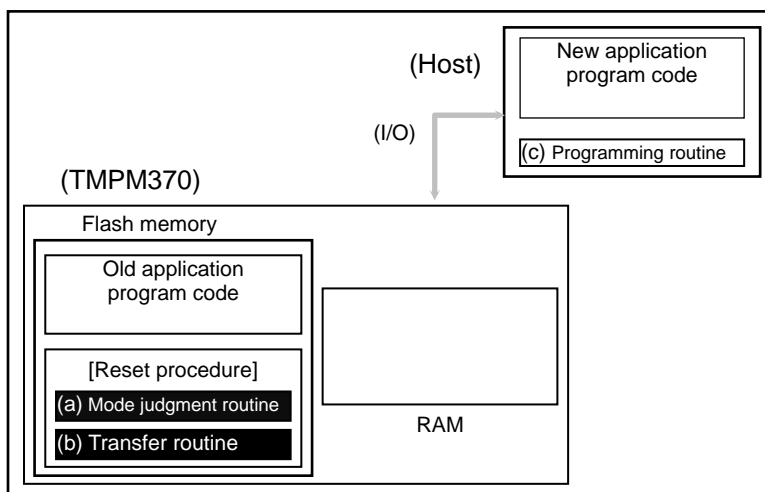
**(Step-1)**

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to transfer new program code. Create hardware and software accordingly. Before installing the TMPM370 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

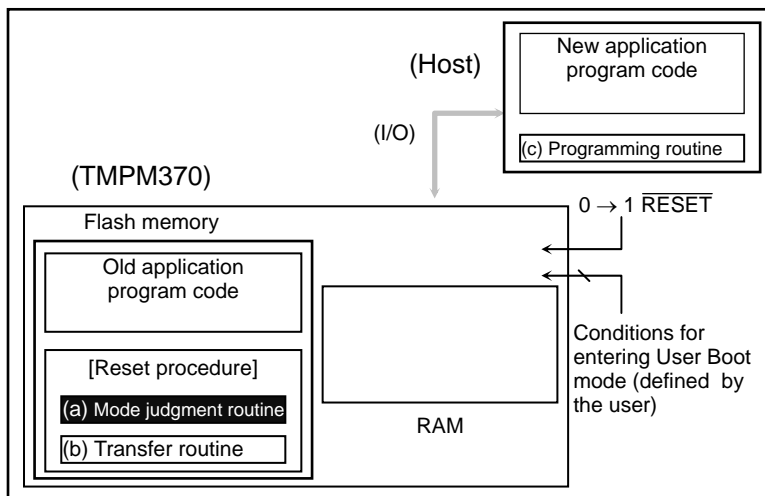
Also, prepare a programming routine shown below on the host controller:

- (c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory



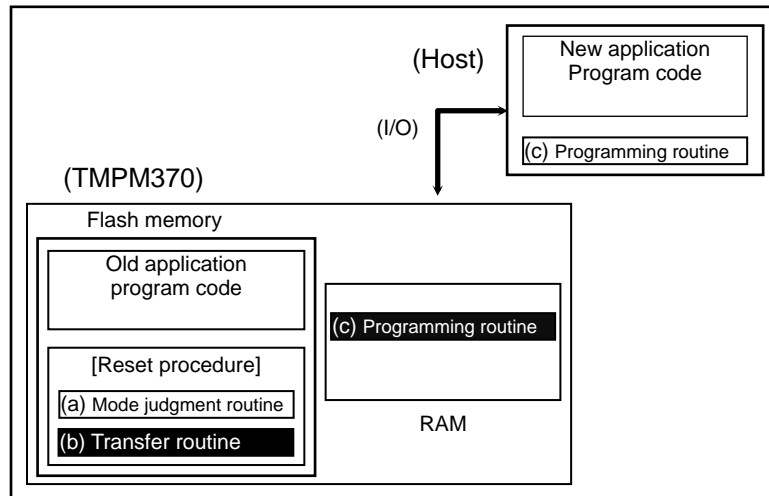
**(Step-2)**

After  $\overline{\text{RESET}}$  is released, the reset procedure determines whether to put the TMPM370 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).



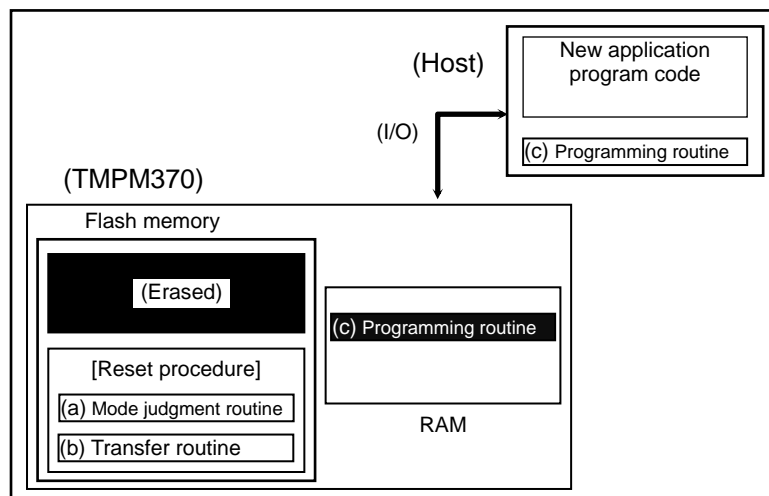
**(Step-3)**

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM370 on-chip RAM.



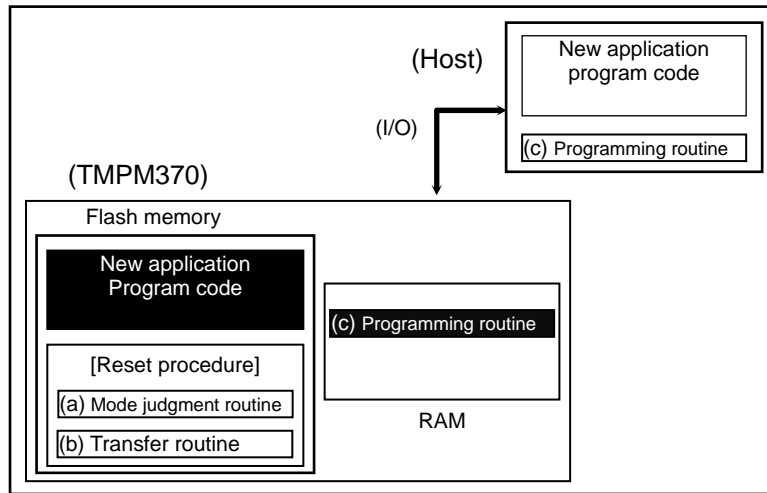
**(Step-4)**

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



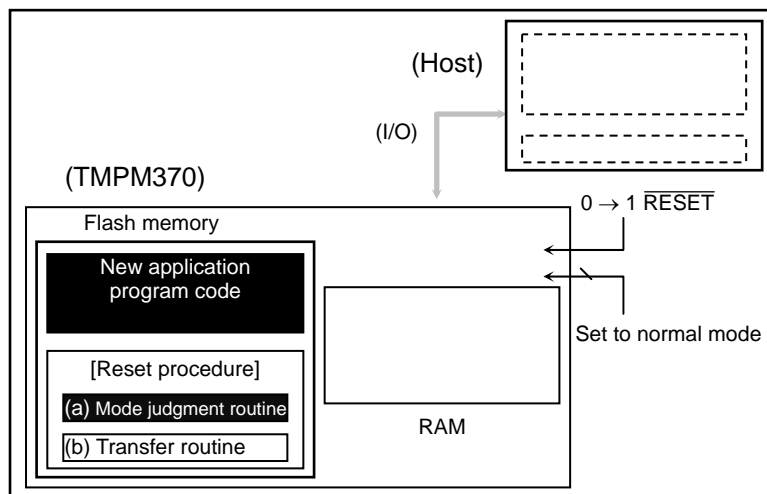
**(Step-5)**

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" low to reset the TMPM370. 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 TMPM370 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 TMPM370 is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM370 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.

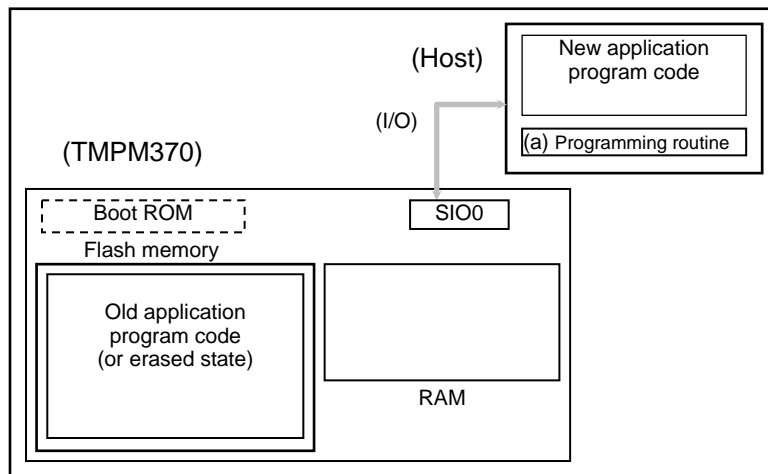
Note : When use Single Boot Mode, it is necessary to reset TMPM370FY from the RESET terminal.

**Single Boot Mode**

(2-A) Using the Program in the On-Chip Boot ROM

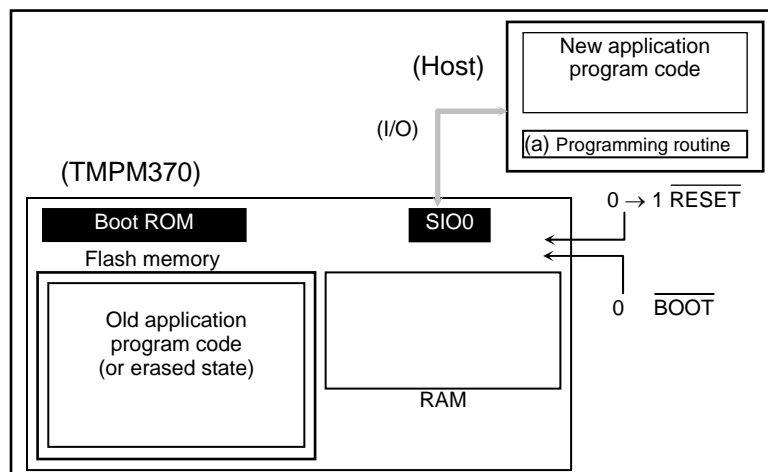
**(Step-1)**

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.



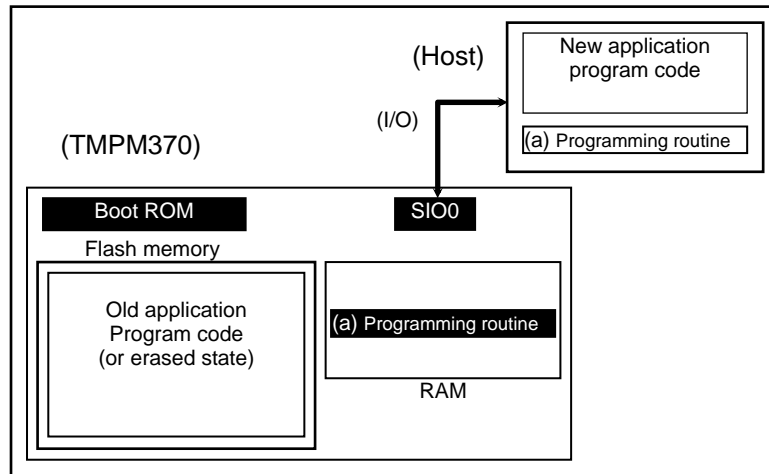
**(Step-2)**

Cancel the reset of the TMPM370 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).



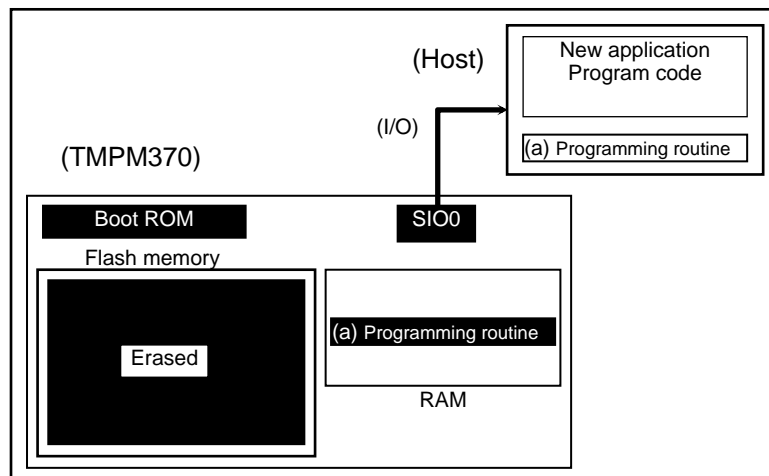
**(Step-3)**

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 TMPM370. The programming routine must be stored in the address range 0x2000\_0400 to the end address of RAM.



**(Step-4)**

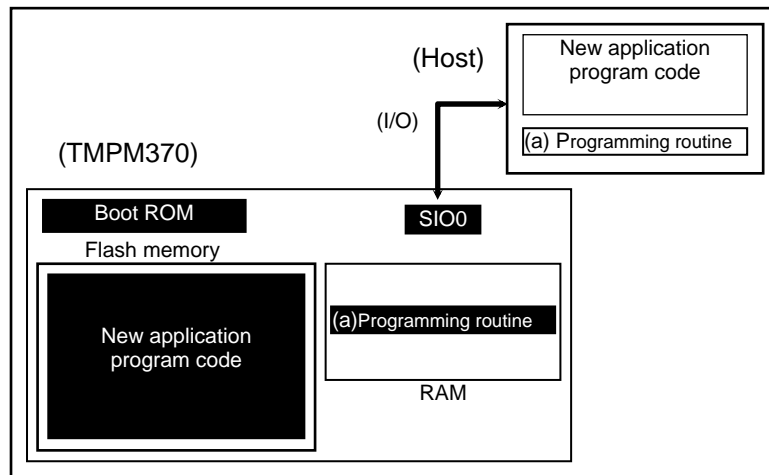
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



**(Step-5)**

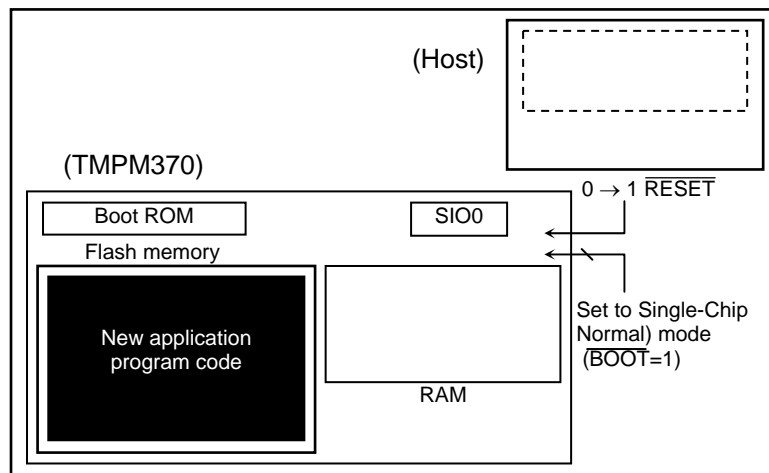
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. Once programming is complete, protection of that flash block is turned on. It is not allowed to move program control from the programming routine (a) back to 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 TMPM370 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 TMPM370 with Single Boot mode following the configuration shown below.

$$\begin{aligned}\overline{\text{BOOT}} \text{ (PF0)} &= 0 \\ \overline{\text{RESET}} &= 0 \rightarrow 1\end{aligned}$$

Set the  $\overline{\text{RESET}}$  input to 0, and set the each  $\overline{\text{BOOT}}$  (PF0) pins to values shown above, and then release RESET (high).

## (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 0x3F83\_FFFF, and the Internal boot ROM (Mask ROM) is mapped to 0x0000\_0000 through 0x0000\_0FFF.

| Product Name | Flash Size | RAM Size | Flash Address<br>(Single Chip/ Single Boot Mode)       | RAM Address               |
|--------------|------------|----------|--|---------------------------|
| TMPM370FY    | 256KB      | 10KB     | 0x0000_0000 - 0x0003_FFFF<br>0x3F80_0000 - 0x3F83_FFFF | 0x2000_0000 - 0x2000_27FF |

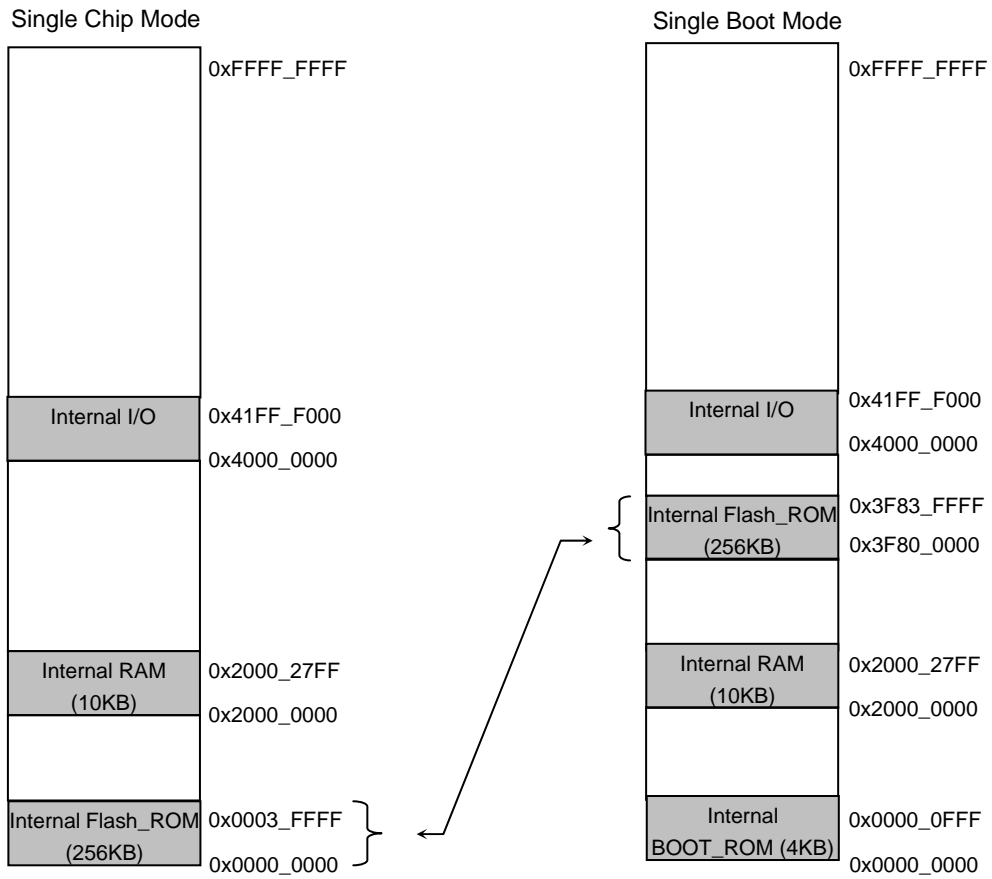


Fig. 20-3 Memory Maps for TPM370FY

## (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  | DVDD5                          | ○         | ○                  |
|                    | AVDD5                          | ○         | ○                  |
|                    | RVDD5                          | ○         | ○                  |
|                    | DVSS                           | ○         | ○                  |
|                    | AVSS                           | ○         | ○                  |
| 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 and Table 20-6 to Table 20-10 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                  |
| 0x20 | Show Flash Memory SUM         |
| 0x30 | Show Product Information      |
| 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.<br>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.<br>TMPM370FY : 0x3F83_FFF0 to 0x3F83_FFFF |

Table 20-6 Transfer Format for the RAM Transfer Command

|          | Byte                            | Data Transferred from the Controller to the TMPM370FY   | Baud rate                         | Data Transferred from the TMPM370FY to the Controller   |
|----------|---------------------------------|---|-----------------------------------|---|
| Boot ROM | 1 byte                          | Serial operation mode and baud rate<br>For UART mode 0x86<br>For I/O Interface mode 0x30                                  | Desired baud rate (Note 1)        | -   |
|          | 2 byte                          | -   |                                   | ACK for the serial operation mode byte<br>For UART mode<br>-Normal acknowledge 0x86<br>(The boot program aborts if the baud rate can not be set correctly.)<br>For I/O Interface mode<br>-Normal acknowledge 0x30 |
|          | 3 byte                          | Command code (0x10)   |                                   | -   |
|          | 4 byte                          | -   |                                   | ACK for the command code byte (Note 2)<br>-Normal acknowledge 0x10<br>-Negative acknowledge 0xN1<br>-Communication error 0xN8   |
|          | 5 byte - 16 byte                | Password sequence (12 bytes)<br>0x3F83_FFF4 to 0x3F83_FFFF  |                                   | -   |
|          | 17 byte                         | Check SUM value for bytes 5 - 16  |                                   | -   |
|          | 18 byte                         | -   |                                   | ACK for the checksum byte (Note 2)<br>-Normal acknowledge 0x10<br>-Negative acknowledge 0xN1<br>-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)<br>-Normal acknowledge 0x10<br>-Negative acknowledge 0xN1<br>-Communication error 0xN8   |
|          | 27 byte<br>~<br>m byte          | RAM storage data  |                                   | -   |
| m + byte | Checksum value for bytes 27 - m | -   |                                   |   |
| m + byte | -                               | ACK for the checksum byte (Note 2)<br>-Normal acknowledge 0x10<br>-Negative acknowledge 0xN1<br>-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 Show Flash Memory Sum Command

|          | Byte   | Data Transferred from the Controller to the TMPM370FY  | Baud rate                  | Data Transferred from the TMPM370FY to the Controller   |
|----------|--------|--|----------------------------|---|
| Boot ROM | 1 byte | Serial operation mode and baud rate<br>For UART mode           0x86<br>For I/O Interface mode   0x30 | Desired baud rate (Note 1) | -   |
|          | 2 byte | -  |                            | ACK for the serial operation mode byte<br>For UART mode<br>-Normal acknowledge       0x86<br>(The boot program aborts if the baud rate can not be set correctly.)<br>For I/O Interface mode<br>-Normal acknowledge       0x30 |
|          | 3 byte | Command code           (0x20)  |                            | -   |
|          | 4 byte | -  |                            | ACK for the command code byte (Note 2)<br>-Normal acknowledge       0x20<br>-Negative acknowledge      0xN1<br>-Communication error      0xN8   |
|          | 5 byte | -  |                            | SUM (upper byte)  |
|          | 6 byte | -  |                            | SUM (lower byte)  |
|          | 7 byte | -  |                            | Checksum value for bytes 5 and 6  |
|          | 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.

Table 20-8 Transfer Format for the Show Product Information Command (1/2)

|                   | Byte              | Data Transferred from the Controller to the TMPM370FY                                    | Baud rate  | Data Transferred from the TMPM370FY to the Controller   |
|-------------------|-------------------|--|--|---|
| Boot ROM          | 1 byte            | Serial operation mode and baud rate<br>For UART mode 0x86<br>For I/O Interface mode 0x30 | Desired baud rate (Note 1)   | -   |
|                   | 2 byte            | -  |  | ACK for the serial operation mode byte<br>For UART mode<br>-Normal acknowledge 0x86<br>(The boot program aborts if the baud rate can not be set correctly.)<br>For I/O Interface mode<br>-Normal acknowledge 0x30 |
|                   | 3 byte            | Command code (0x30)  |  | -   |
|                   | 4 byte            | -  |  | ACK for the command code byte (Note 2)<br>-Normal acknowledge 0x30<br>-Negative acknowledge 0xN1<br>-Communication error 0xN8   |
|                   | 5 byte            | -  |  | Flash memory data at address 0x3F83_FFF0  |
|                   | 6 byte            | -  |  | Flash memory data at address 0x3F83_FFF1  |
|                   | 7 byte            | -  |  | Flash memory data at address 0x3F83_FFF2  |
|                   | 8 byte            | -  |  | Flash memory data at address 0x3F83_FFF3  |
|                   | 9 byte - 20 byte  | -  |  | Product name (12-byte ASCII code)<br>From the 9th byte: 'TMPM370FY[_]_'<br>(where [ ] is a space)   |
|                   | 21 byte - 24 byte | -  |  | Password comparison start address (4 bytes)<br>From the 21 <sup>st</sup> byte: 0xF4, 0xFF, 0x83, 0x3F   |
|                   | 25 byte - 28 byte | -  |  | RAM start address (4 bytes)<br>0x00, 0x00, 0x00 and 0x20 from the 25 <sup>th</sup> byte   |
|                   | 29 byte - 32 byte | -  |  | Dummy data (4 bytes)<br>0x00, 0x00, 0x00 and 0x00 from the 29 <sup>th</sup> byte  |
|                   | 33 byte - 36 byte | -  |  | RAM end address (4 bytes)<br>From the 33 <sup>rd</sup> byte: 0xFF, 0x27, 0x00, 0x20<br>(Note 3)   |
|                   | 37 byte - 40 byte | -  |  | Dummy data (4 bytes)<br>0x00, 0x00, 0x00 and 0x00 from the 37 <sup>th</sup> byte.   |
|                   | 41 byte - 44 byte | -  |  | Dummy data (4 bytes)<br>0x00, 0x00, 0x00 and 0x00 from the 41 <sup>st</sup> byte  |
|                   | 45 byte - 46 byte | -  |  | Fuse information (2 bytes)<br>0x00 and 0x00 from the 45 <sup>th</sup> byte.   |
|                   | 47 byte - 50 byte | -  |  | Flash memory start address (4 bytes)<br>0x00, 0x00, 0x80 and 0x3F from the 47 <sup>th</sup> byte  |
| 51 byte - 54 byte | -                 |  | Flash memory end address (4 bytes)<br>From the 51 <sup>st</sup> byte: 0xFF, 0xFF, 0x83, 0x3F<br>(Note 4) |   |
| 55 byte - 56 byte | -                 |  | Flash memory block count (2 bytes)<br>From the 55 <sup>th</sup> byte: 0x06, 0x00                         |   |

Table 20-8 Transfer Format for the Show Product Information Command (2/2)

|          | Byte                 | Data Transferred from the Controller to the TMPM370FY | Baud rate | Data Transferred from the TMPM370FY to the Controller  |
|----------|----------------------|---|-----------|--|
| Boot ROM | 57 byte -<br>60 byte | -   |           | Start address of a group of the same-size (16K) flash blocks (4 bytes)<br>From 57 <sup>th</sup> byte: 0x00, 0x00, 0x80, 0x3F   |
|          | 61 byte -<br>64 byte | -   |           | Size (in halfwords) of the same-size (16K) flash blocks (4 bytes)<br>0x00, 0x20, 0x00 and 0x00 from the 61 <sup>st</sup> byte  |
|          | 65 byte              | -   |           | Number of flash blocks of the same size (16K) (1 byte)<br>0x02   |
|          | 66 byte -<br>69 byte | -   |           | Start address of a group of the same-size (32K) flash blocks (4 bytes)<br>From 66 <sup>th</sup> byte: 0x00, 0x80, 0x80, 0x3F   |
|          | 70 byte -<br>73 byte |   |           | Size (in halfwords) of the same-size (32K) flash blocks (4 bytes)<br>0x00, 0x40, 0x00 and 0x00 from the 70 <sup>th</sup> byte  |
|          | 74 byte              |   |           | Number of flash blocks of the same size (32K) (1 byte)<br>0x01   |
|          | 75 byte -<br>78 byte |   |           | Start address of a group of the same-size (64K) flash blocks (4 bytes)<br>0x00, 0x00, 0x81 and 0x3F from 75 <sup>th</sup> byte |
|          | 79 byte -<br>82 byte |   |           | Size (in halfwords) of the same-size (64K) flash blocks (4 bytes)<br>0x00, 0x80, 0x00 and 0x00 from the 79 <sup>th</sup> byte  |
|          | 83 byte              |   |           | Number of flash blocks of the same size (64K) (1 byte)<br>From 83 <sup>rd</sup> byte: 0x03                                     |
|          | 84 byte -<br>87 byte |   |           | Start address of a group of the same-size (128K) flash blocks (4 bytes)<br>From 84 <sup>th</sup> byte: 0x00, 0x00, 0x82, 0x3F  |
|          | 88 byte -<br>91 byte |   |           | Size (in halfwords) of the same-size (128K) flash blocks (4 bytes)<br>0x00, 0x00, 0x00 and 0x00 from the 88 <sup>th</sup> byte |
|          | 92 byte              |   |           | Number of flash blocks of the same size (128K) (1 byte)<br>0x00 (Note 5)   |
|          | 93 byte              |   |           | Checksum value for bytes 5 - 92  |
|          | 94 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.



Table 20-9 Transfer Format for the Chip and Protection Bit Erase Command

|          | Byte   | Data Transferred from the Controller to the TMPM370FY  | Baud rate                  | Data Transferred from the TMPM370FY to the Controller   |
|----------|--------|--|----------------------------|---|
| Boot ROM | 1 byte | Serial operation mode and baud rate<br>For UART mode           0x86<br>For I/O Interface mode   0x30 | Desired baud rate (Note 1) | —   |
|          | 2 byte | —  |                            | ACK for the serial operation mode byte<br>For UART mode<br>-Normal acknowledge       0x86<br>For I/O Interface mode<br>-Normal acknowledge       0x30<br>(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)<br>-Normal acknowledge       0x40<br>-Negative acknowledge      0xN1<br>-Communication error       0xN8  |
|          | 5 byte | Chip erase command code   (0x54)   |                            | —   |
|          | 6 byte | —  |                            | ACK for the command code byte (Note 2)<br>-Normal acknowledge       0x54<br>-Negative acknowledge      0xN1<br>-Communication error       0xN8  |
|          | 7 byte | —  |                            | ACK for the chip erase command code byte<br><br>-Normal acknowledge       0x4F<br>-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 three 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.

## 2. Show Flash Memory Sum command

The Show Flash Memory Sum command adds the entire contents of the flash memory together. The boot program does not provide a command to read out the contents of the flash memory. Instead, the Flash Memory Sum command can be used for software revision management.

## 3. Show Product Information command

The Show Product Information command provides the product name, on-chip memory configuration and the like. This command also reads out the contents of the flash memory locations at addresses shown below. In addition to the Show Flash Memory Sum command, these locations can be used for software revision management.

| Product name | Area        |             |
|--------------|-------------|-------------|
| TMPM370FY    | 0x3F83_FFF0 | 0x3F87_FFF3 |

## 4. Chip and Protection Bit Erase command

This command erases the entire area of the flash memory automatically without verifying a password. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. When the command is completed, the FCSECBIT <SECBIT> bit is set to "1".

This command serves to recover boot programming operation when a user forgets the password. Therefore password verification is not executed.

- 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 desired 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 5<sup>th</sup> 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        |             |
|--------------|-------------|-------------|
| TMPM370FY    | 0x3F83_FFF4 | 0x3F83_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 TMPM370FY. 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) Show Flash Memory Sum 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 Flash Memory Sum command is 20H.
  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. 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 Show Flash Memory Sum command was received, the boot program echoes back a value of 0x20 and then branches to the Show Flash Memory Sum routine. If the 3rd byte is not a valid command, the boot program sends back 0xN1 (bit 0) to the controller and returns to the command wait state (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 Show Flash Memory Sum routine adds all the bytes of the flash memory together. The 5th and 6th bytes, transmitted from the target board to the controller, indicate the upper and lower bytes of this total sum, respectively. For details on sum calculation, see a later section "Calculation of the Show Flash Memory Sum Command".
5. The 7th byte is a checksum value for the 5th and 6th bytes. To calculate the checksum value, add the 5th and 6th bytes together, drop the carry and take the two's complement of the sum. Transmit this checksum value from the controller to the target board.
6. The 8th byte is the next command code.



- 3) Show Product Information Command (See Table 20-8)
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 30H.
  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 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. 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 Show Flash Memory Sum command was received, the boot program echoes back a value of 0x30 and then branches to the Show Flash Memory Sum routine. 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 to 8th bytes, transmitted from the target board to the controller, are the data read from addresses shown below in the flash memory. Software version management is possible by storing a software ID in these locations.

| Product name | Area        |             |
|--------------|-------------|-------------|
| TMPM370FY    | 0x3F87_FFF0 | 0x3F87_FFF3 |

5. The 9th to 20th bytes, transmitted from the target board to the controller, indicate the product name as shown below (where [ ] is a space) in ASCII code.

| Product name | Code                       |
|--------------|----------------------------|
| TMPM370FY    | T,M,P,M,3,7,0,F,Y,_, [ ],_ |

6. The 21st to 24th bytes, transmitted from the target board to the controller, indicate the start address of the flash memory area containing the password. Each product has own start address shown below.

| Product name | Address                |
|--------------|------------------------|
| TMPM370FY    | 0xF4, 0xFF, 0x87, 0x3F |

7. The 25th to 28th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip RAM, are 0x00, 0x00, 0x00 and 0x20.
8. The 29th to 32nd bytes, transmitted from the target board to the controller, are dummy data (0x00, 0x00, 0x00 and 0x00).
9. The 33rd to 36th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip RAM, are 0xFF, 0x27, 0x00 and 0x20.
10. The 37th to 40th bytes, transmitted from the target board to the controller, are 0x00, 0x00, 0x00 and 0x00. The 41st to 44th bytes, transmitted from the target board to the controller, are 0x00, 0x00, 0x00 and 0x00.
11. The 45th and 46th bytes transmitted are 0x00, 0x00.
12. The 47th to 50th bytes, transmitted from the target board to the controller, indicate the start address of the on-chip flash memory, are 0x00, 0x00, 0x80, and 0x3F.
13. The 51st to 54th bytes, transmitted from the target board to the controller, indicate the end address of the on-chip flash memory, are 0xFF, 0xFF, 0x83 and 0x3F.
14. The 55th to 56th bytes, transmitted from the target board to the controller, indicate the number of flash blocks available, are 0x06 and 0x00.
15. The 57th to 83rd bytes, transmitted from the target board to the controller, contain information about the flash blocks. Flash blocks of the same size are treated as a group. Information about the flash blocks indicate the start address of a group, the size of the blocks in that group (in halfwords) and the number of the blocks in that group.  
  
The 57th to 65th bytes are the information about the 16-kbyte blocks. The 66th to 74th bytes are the information about the 32-kbyte blocks. The 75th to 83rd bytes are the information about the 64-kbyte blocks. The 84th to 92nd bytes are the information about the 128-kbyte blocks. See Table 20-8 for the values of bytes transmitted.
16. The 93rd byte, transmitted from the target board to the controller, is a checksum value for the 5th to 92nd bytes. The checksum value is calculated by adding all these bytes together, dropping the carry and taking the two's complement of the total sum.
17. The 94th byte is the next command code.

- 4) Chip and Protection Bit Erase command (See Table 20-9)
  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.  
  
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.
  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 7<sup>th</sup> 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.

## 5) Acknowledge Responses

The boot program represents processing states with specific codes. Table 20-10 to Table 20-13 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 20-10 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 20-11 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.                                      |
| 0x20            | The Show Flash Memory Sum command was received.                             |
| 0x30            | The Show Product Information 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 20-12 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.              |

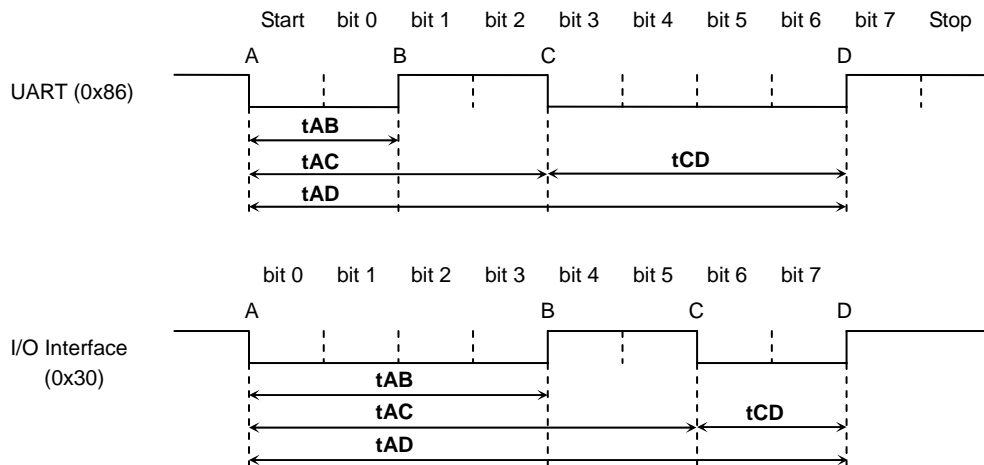
**(Note)** The upper four bits of the ACK response are the same as those of the operation command code. It is 1 ( N = RAM transfer command data [7:4] ) when password error occurs.

**Table 20-13 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. |

## 6) 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  $t_{AB}$ ,  $t_{AC}$  and  $t_{AD}$ . Fig. 20-5 shows a flowchart describing the steps to determine the intervals of  $t_{AB}$ ,  $t_{AC}$  and  $t_{AD}$ . 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  $t_{AB}$ ,  $t_{AC}$  and  $t_{AD}$  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  $t_{AB}$  is equal to or less than the length of  $t_{CD}$ , the serial operation mode is determined as UART mode. If the length of  $t_{AB}$  is greater than the length of  $t_{CD}$ , 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  $t_{AB}$  is

greater than  $t_{CD}$  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  $t_{AB}$  is greater than  $t_{CD}$  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).

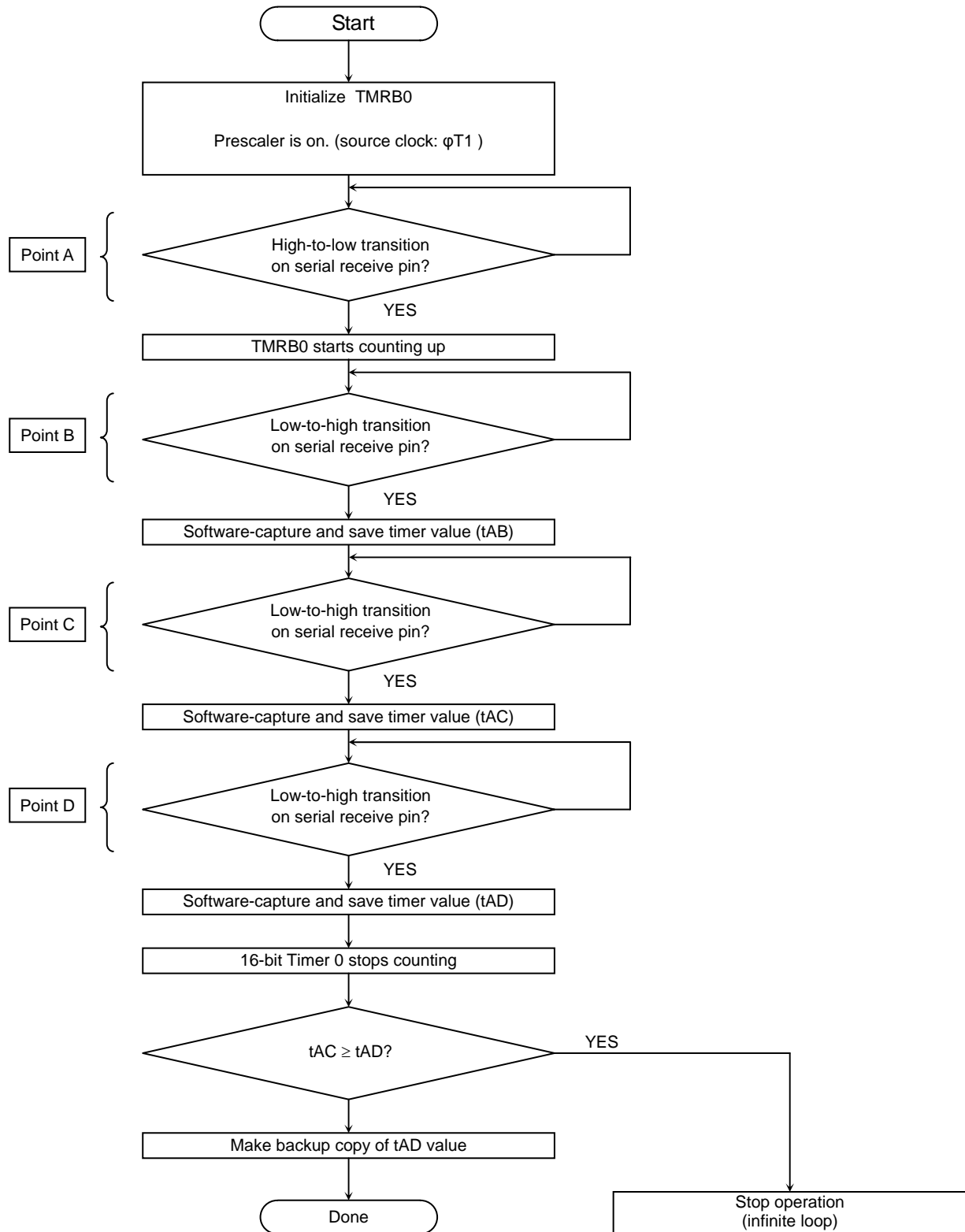


Fig. 20-5 Serial Operation Mode Byte Reception Flow

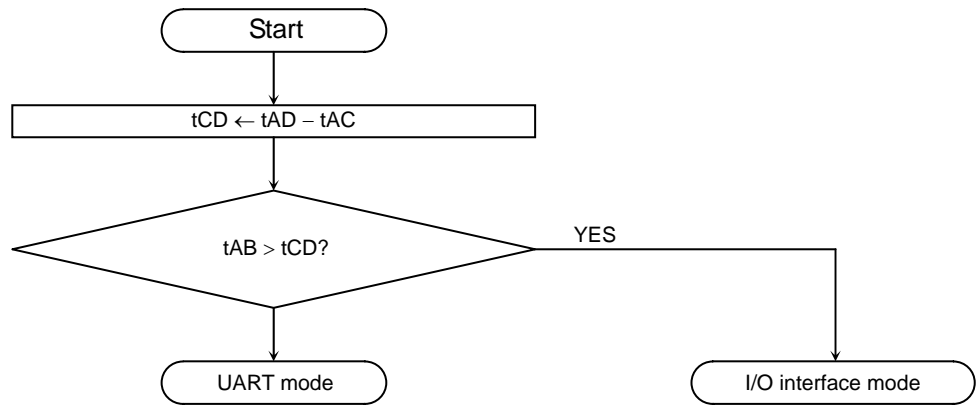


Fig. 20-6 Serial Operation Mode Determination Flow

7) 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        |             |
|--------------|-------------|-------------|
| TMPM370FY    | 0x3F83_FFF4 | 0x3F83_FFFF |

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.

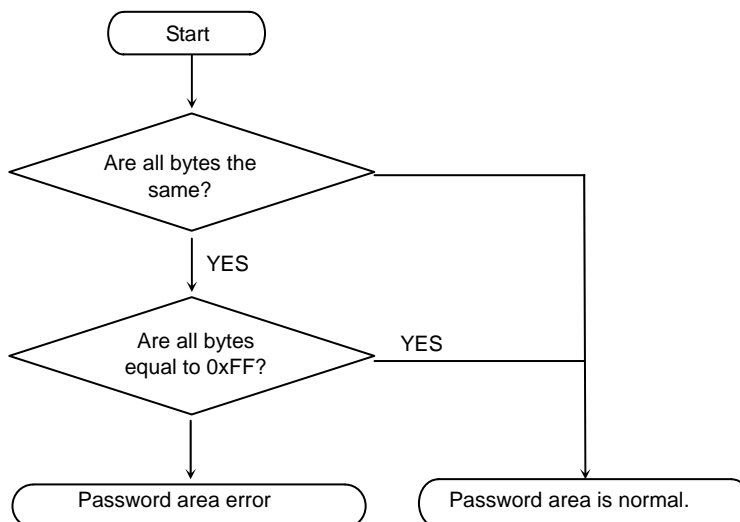


Fig. 20-7 Password Area Verification Flow



## 8) Calculation of the Show Flash Memory Sum Command

The result of the sum calculation “byte + byte + byte + ...” is responded by a word quantity. The Show Flash Memory Sum command adds all 256 Kbytes of the flash memory together and provides the total sum as a word quantity. The sum is sent to the controller, with the upper eight bits first, followed by the lower eight bits.

|          |      |  |
|----------|------|--|
| Example) |      | <p>For the interest of simplicity, assume the depth of the flash memory is four locations. Then the sum of the four bytes is calculated as:</p> $0xA1 + 0xB2 + 0xC3 + 0xD4 = 0x02EA$ <p>Hence, 0x02 is first sent to the controller, followed by 0xEA.</p> |
|          | 0xA1 |  |
|          | 0xB2 |  |
|          | 0xC3 |  |
|          | 0xD4 |  |

## 9) 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.

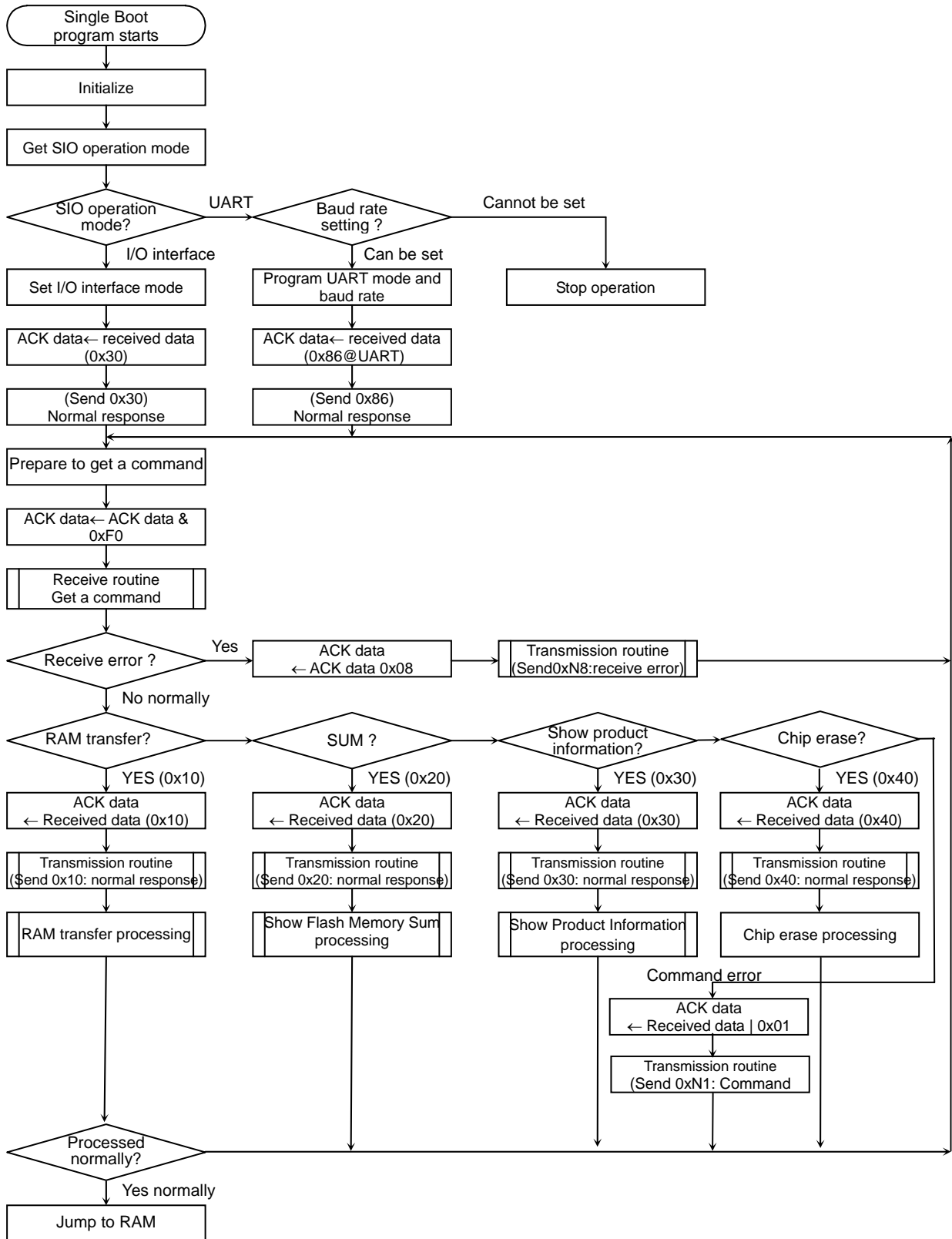


Fig. 20-8 Overall Boot Program Flow

### 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

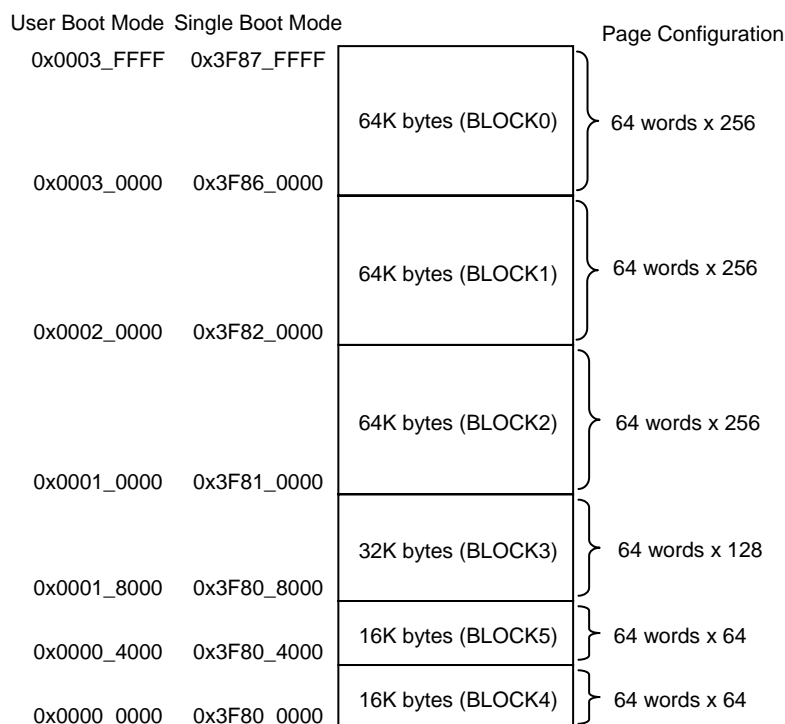
Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands. In writing or erasing, use 32-bit data transfer command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

**Table 20-14 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



**Fig. 20-9 Block Configuration of Flash Memory (TMPM370FY)**

## (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 FCFLCS 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 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 TMPM370FY 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 FCFLCS [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 FCFLCS [0] <RDY/BSY> (See Table 20-15). 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 FCFLCS [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 FCFLCS [0] <RDY/BSY> (Table 20-15). If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

**(Note) Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.**

2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FCFLCS [0] <RDY/BSY> (See Table 20-15). 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 FCFLCS <RDY/BSY> (See Table 20-15). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If

it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

4) Automatic programming of protection bits (for each block)

This device is implemented with protection bits. This protection can be set for each block. See Table 20-20 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 FCFLCS <BLPRO> register to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY/BSY> (See Table 20-15). 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 FCFLCS <BLPRO> bits are set to "1" indicating that it is in the protected state (See Table 20-15). 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 FCFLCS <BLPRO> whether all the <BLPRO> bits are set to "1" or not if FCSECBIT<SECBIT> is 0x1. Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See chapter 17 for details.

• **When all the FCFLCS <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 FCFLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FCFLCS will be set to "0x00000001." While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FCFLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

• **When the FCFLCS <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 20-21. 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 FCFLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FCFLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FCFLCS <BLPRO> selected for erasure are set to "0."

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

**(Note) The 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 register is used to monitor the status of the flash memory and to indicate the protection status of each block.

**Table 20-15 Flash Control Register**

FCFLCS  
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  | -           | -  | BLPRO5  | BLPRO4  | BLPRO3  | BLPRO2  | BLPRO1  | BLPRO0   |
| Read/Write  | R           |    | R   | R   | R   | R   | R   | R  |
| After reset | 0           |    | ( Note 2 )  | ( Note 2 )  | ( Note 2 )  | ( Note 2 )  | ( Note 2 )  | ( Note 2 )   |
| Function    | "0" is read |    | Protection for Block 5<br>0: disabled<br>1: enabled | Protection for Block 4<br>0: disabled<br>1: enabled | Protection for Block 3<br>0: disabled<br>1: enabled | Protection for Block 2<br>0: disabled<br>1: enabled | Protection for Block 1<br>0: disabled<br>1: enabled | Protection for Block 0<br>0: disabled<br>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)<br>0: Auto operating<br>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 [21:16]: 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 20-16 Security bit register**

FCSECBIT  
0x41FF\_F010

|             | 31        | 30 | 29 | 28 | 27 | 26 | 25 | 24                                       |
|-------------|-----------|----|----|----|----|----|----|--|
| bit Symbol  | -         | -  | -  | -  | -  | -  | -  | -  |
| Read/Write  | R         |    |    |    |    |    |    |  |
| After reset | 0         |    |    |    |    |    |    |  |
| Function    | '0'isread |    |    |    |    |    |    |  |
|             | 23        | 22 | 21 | 20 | 19 | 18 | 17 | 16                                       |
| bit Symbol  | -         | -  | -  | -  | -  | -  | -  | -  |
| Read/Write  | R         |    |    |    |    |    |    |  |
| After reset | 0         |    |    |    |    |    |    |  |
| Function    | '0'isread |    |    |    |    |    |    |  |
|             | 15        | 14 | 13 | 12 | 11 | 10 | 9  | 8  |
| bit Symbol  | -         | -  | -  | -  | -  | -  | -  | -  |
| Read/Write  | R         |    |    |    |    |    |    |  |
| After reset | 0         |    |    |    |    |    |    |  |
| Function    | '0'isread |    |    |    |    |    |    |  |
|             | 7         | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| bit Symbol  | -         | -  | -  | -  | -  | -  | -  | SECBIT                                   |
| Read/Write  | R         |    |    |    |    |    |    | R/W                                      |
| After reset | 0         |    |    |    |    |    |    | 1  |
| Function    | '0'isread |    |    |    |    |    |    | Security bits<br>0:disabled<br>1:enabled |

**(Note)** This register is initialized only by power-on reset.

## (6) List of Command Sequences

Table 20-17 Flash Memory Access from the Internal CPU

| Command sequence           | First bus cycle | Second bus cycle | Third bus cycle | Fourth bus cycle | Fifth bus cycle | Sixth bus cycle | Seventh bus cycle |
|----------------------------|-----------------|------------------|-----------------|------------------|-----------------|-----------------|-------------------|
|                            | Addr.           | Addr.            | Addr.           | Addr.            | Addr.           | Addr.           | Addr.             |
|                            | Data            | Data             | Data            | Data             | Data            | Data            | Data              |
| Read                       | 0xXX            | -                | -               | -                | -               | -               | -                 |
|                            | 0xF0            | -                | -               | -                | -               | -               | -                 |
| Read/Reset                 | 0x54XX          | 0xAAXX           | 0x54XX          | RA               | -               | -               | -                 |
|                            | 0xAA            | 0x55             | 0xF0            | RD               | -               | -               | -                 |
| ID-Read                    | 0x54XX          | 0xAAXX           | 0x54XX          | IA               | 0xXX            | -               | -                 |
|                            | 0xAA            | 0x55             | 0x90            | 0x00             | ID              | -               | -                 |
| Automatic page programming | 0x54XX          | 0xAAXX           | 0x54XX          | PA               | PA              | PA              | PA                |
|                            | 0xAA            | 0x55             | 0xA0            | PD0              | PD1             | PD2             | PD3               |
| Automatic chip erase       | 0x54XX          | 0xAAXX           | 0x54XX          | 0x54XX           | 0xAAXX          | 0x54XX          | -                 |
|                            | 0xAA            | 0x55             | 0x80            | 0xAA             | 0x55            | 0x10            | -                 |
| Auto Block erase           | 0x54XX          | 0xAAXX           | 0x54XX          | 0x54XX           | 0xAAXX          | BA              | -                 |
|                            | 0xAA            | 0x55             | 0x80            | 0xAA             | 0x55            | 0x30            | -                 |
| Protection bit programming | 0x54XX          | 0xAAXX           | 0x54XX          | 0x54XX           | 0xAAXX          | 0x54XX          | PBA               |
|                            | 0xAA            | 0x55             | 0x9A            | 0xAA             | 0x55            | 0x9A            | 0x9A              |
| Protection bit erase       | 0x54XX          | 0xAAXX           | 0x54XX          | 0x54XX           | 0xAAXX          | 0x54XX          | PBA               |
|                            | 0xAA            | 0x55             | 0x6A            | 0xAA             | 0x55            | 0x6A            | 0x6A              |

## Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address  
PD: Program data (32 bit data)  
After 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 20-18 Address Bit Configuration for Bus Write Cycles**

| Address                    | Addr [31:19]  | Addr [18]                              | Addr [17] | Addr [16]           | Addr [15]     | Addr [14]                                      | Addr [13:11] | Addr [10]                              | Addr [9]  | Addr [8]  | Addr [7:0] |
|----------------------------|---|--|-----------|---------------------|---------------|--|--------------|--|---|---|------------|
| Normal commands            | <b>Normal bus write cycle address configuration</b>   |  |           |                     |               |  |              |  |   |   |            |
|                            | Flash area  | "0" is recommended.                    |           |                     |               | Command  |              |  |   | Addr[1:0]="0" (fixed)<br>Others:0 (recommended) |            |
| ID -READ                   | <b>IA: ID address (Set the fourth bus write cycle address for ID-Read operation)</b>                    |  |           |                     |               |  |              |  |   |   |            |
|                            | Flash area  | "0" is recommended.                    |           | ID address          |               | Addr[1:0]="0" (fixed) , Others:0 (recommended) |              |  |   |   |            |
| Block erase                | <b>BA: Block address (Set the sixth bus write cycle address for block erase operation)</b>              |  |           |                     |               |  |              |  |   |   |            |
|                            | Block selection (Table 20-19)   |  |           |                     |               | Addr[1:0]="0" (fixed) , Others:0 (recommended) |              |  |   |   |            |
| Auto page programming      | <b>PA: Program page address (Set the fourth bus write cycle address for page programming operation)</b> |  |           |                     |               |  |              |  |   |   |            |
|                            | Page selection  |  |           |                     |               |  |              |  | Addr[1:0]="0" (fixed)<br>Others:0 (recommended) |   |            |
| Protection bit programming | <b>PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)</b> |  |           |                     |               |  |              |  |   |   |            |
|                            | Flash area  | Protection bit selection (Table 20-20) |           |                     | Fixed to "0". |  |              | Protection bit selection (Table 20-20) |   | Addr[1:0]="0" (fixed)<br>Others:0 (recommended) |            |
| Protection bit erase       | <b>PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)</b> |  |           |                     |               |  |              |  |   |   |            |
|                            | Flash area  | Protection bit selection (Table 20-21) |           | "0" is recommended. |               |  |              | Protection bit selection (Table 20-21) |   | Addr[1:0]="0" (fixed)<br>Others:0 (recommended) |            |
| Block erase                | <b>BA: Block address (Set the sixth bus write cycle address for block erase operation)</b>              |  |           |                     |               |  |              |  |   |   |            |
|                            | Block selection (Table 20-19)   |  |           |                     |               | Addr[1:0]="0" (fixed) , Others:0 (recommended) |              |  |   |   |            |

- (Note 1)** Table 20-17 "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 20-19 Block Address Table

| Block | Address<br>(User boot mode) | Address<br>(Single boot mode) | Size<br>(Kbyte) |
|-------|-----------------------------|-------------------------------|-----------------|
| 4     | 0x0000_0000-0x0000_3FFF     | 0x3F80_0000-0x3F80_3FFF       | 16              |
| 5     | 0x0000_4000-0x0000_7FFF     | 0x3F80_4000-0x3F80_7FFF       | 16              |
| 3     | 0x0000_8000-0x0000_FFFF     | 0x3F80_8000-0x3F80_FFFF       | 32              |
| 2     | 0x0001_0000-0x0001_FFFF     | 0x3F81_0000-0x3F81_FFFF       | 64              |
| 1     | 0x0002_0000-0x0002_FFFF     | 0x3F82_0000-0x3F82_FFFF       | 64              |
| 0     | 0x0003_0000-0x0003_FFFF     | 0x3F83_8000-0x3F83_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 20-20 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] |
| <b>[TMPM370FY]</b> |                |                                     |              |               |                 |              |             |             |
| 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           |
| Block4             | BLPRO4         | 0                                   | 1            |               |                 |              | 0           | 0           |
| Block5             | BLPRO5         | 0                                   | 1            |               |                 |              | 0           | 1           |

Table 20-21 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            |
| Block4 ~ 5 | BLPRO4 ~ 5     | 0   | 1            |

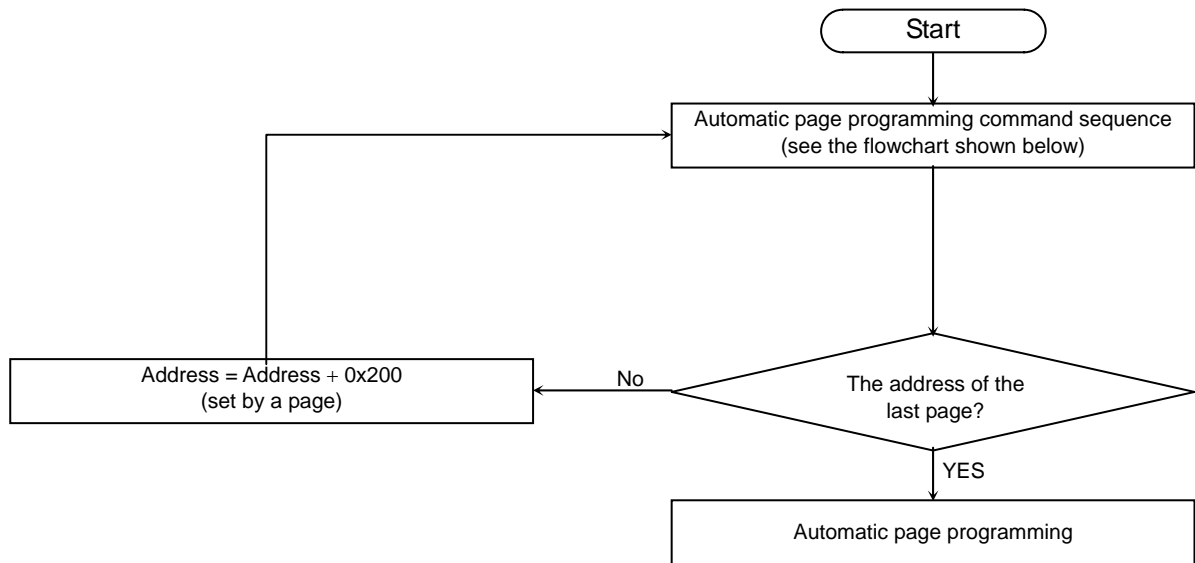
**(Note)** The protection bit erase command cannot erase by individual block.

Table 20-22 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)

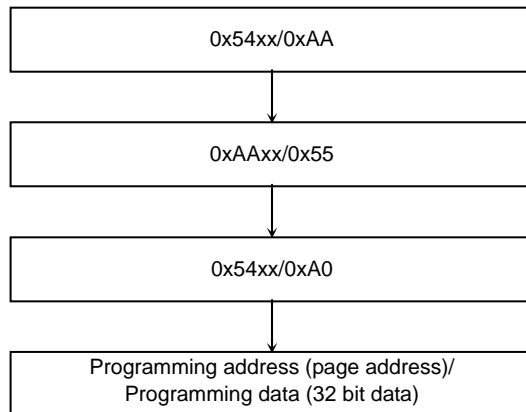
| IA [15:14] | ID [7: 0 ] | Code              |
|------------|------------|-------------------|
| 00b        | 0x98       | Manufacturer code |
| 01b        | 0x5A       | Device code       |
| 10b        | Reserved   | ---               |
| 11b        | 0x13       | Macro code        |



(8) Flowchart



Automatic Page Programming Command Sequence (Address/ Command)



(Note) Command sequence is executed by 0x54xx or 0x55xx.

Fig. 20-10 Automatic Programming

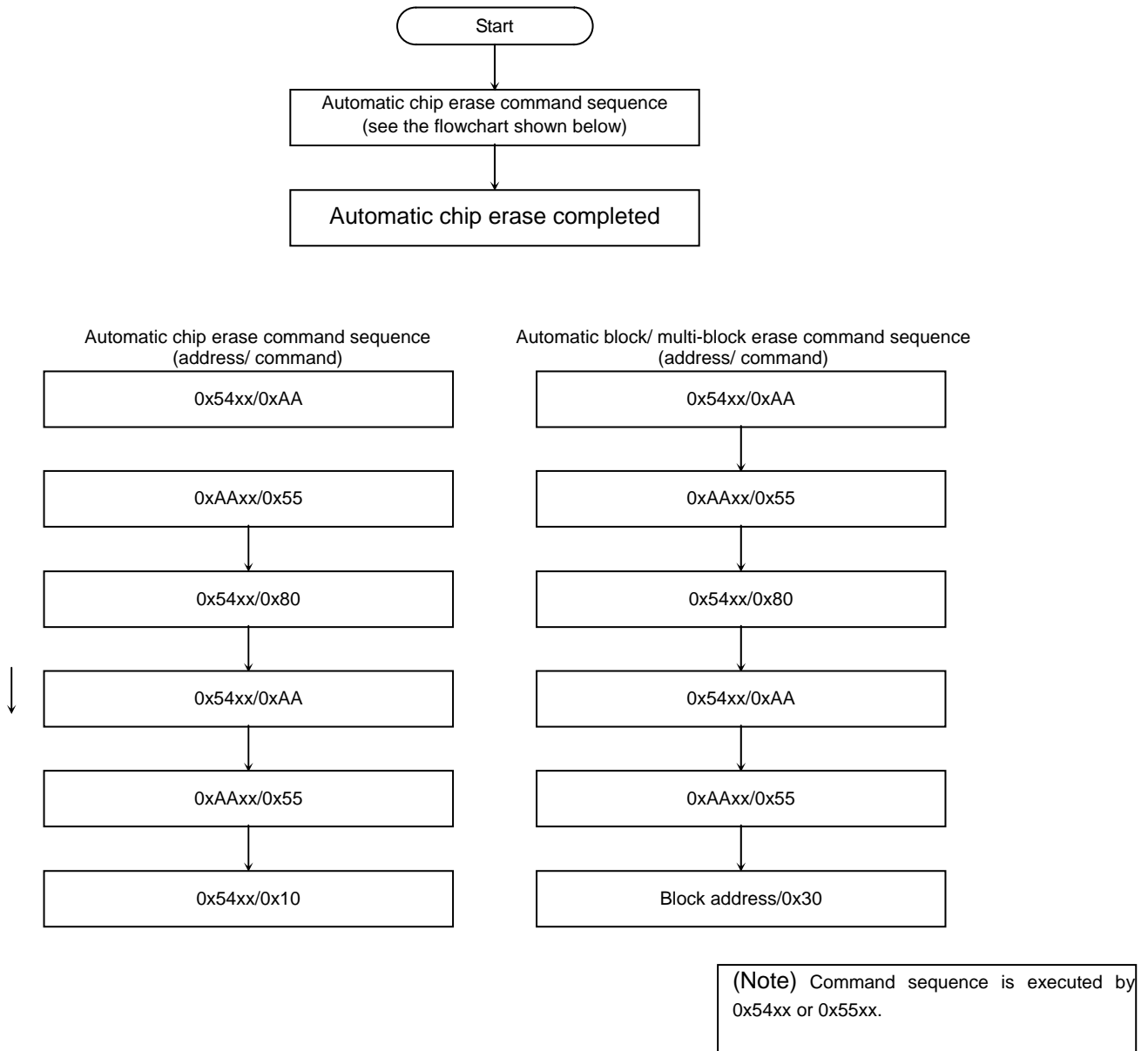


Fig. 20-11 Automatic Erase

## 21 Protect/security function

### 21.1 OutLine

The TPM370FY 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         | Data in the ROM area cannot be read out when writer mode is set. By executing readout, the company code 0x0098 is read. The ROM reading operation is available with a single chip mode and single boot mode. |
| 2) Debug port               | Communication of JTAG/SW and trace are prohibited.   |
| 3) Command for flash memory | Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/erase-protection will erase all the contents of flash memory include protection bits.   |

### 21.3 Resistors

The flash control register shows the status of the flash memory operation and the protection of each block.

Table 21-2 Flash control register

FCFLCS  
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  | -                   | -  | BLPRO5   | BLPRO4   | BLPRO3   | BLPRO2   | BLPRO1   | BLPRO0   |
| Read/Write  | R                   |    | R  | R  | R  | R  | R  | R  |
| after Reset | 0                   |    | (note2)  | (note2)  | (note2)  | (note2)  | (note2)  | (note2)  |
| function    | Reading data is '0' |    | Block5 protect status<br>0: no protect status.<br>1: protect status. | Block4 protect status<br>0: no protect status.<br>1: protect status. | Block3 protect status<br>0: no protect status.<br>1: protect status. | Block2 protect status<br>0: no protect status.<br>1: protect status. | Block1 protect status<br>0: no protect status.<br>1: protect status. | Block0 protect status<br>0: no protect status.<br>1: protect status.           |
|             | 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  | -                   | -  | -  | -  | -  | -  | -  | RDY/BSY  |
| Read/Write  | R                   |    |  |  |  |  |  | R  |
| after reset | 0                   |    |  |  |  |  |  | 1  |
| Function    | reading data is '0' |    |  |  |  |  |  | Ready/Busy (note1)<br>0: under auto operation<br>1: auto 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 [21:16]: Protection status bits**

Each of the protection bits (6 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 register

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<br>0: disable<br>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.

Erasing of the protection bits is done by two groups of the blocks: block 0 through 3 and block 4 through 5. 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 erase all the contents of flash memory include protection bits. To write and erase the protection bits, command sequence is used.

See chapter 18 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] OP-amp (AMP), Comparator (CMP)
- [7] Watchdog timer (WDT)
- [8] Clock generator (CG)
- [9] Oscillation frequency detector (OFD)
- [10] Power on reset (POR), Voltage detecting circuit (VLTD)
- [11] Vector engine (VE)
- [12] Programmable motor driver (PMD)
- [13] Flash
- [14] 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 [1/5]

#### <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 |               | 0x4000_0078 | PBIE          |
| 0x4000_0049 | <R0>          | 0x4000_0059 |               | 0x4000_0069 |               | 0x4000_0079 | <R0>          |
| 0x4000_004A | <R0>          | 0x4000_005A |               | 0x4000_006A |               | 0x4000_007A | <R0>          |
| 0x4000_004B | <R0>          | 0x4000_005B |               | 0x4000_006B |               | 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 C>

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4000_0080 | PCDATA        | 0x4000_0090 |               | 0x4000_00A0 |               | 0x4000_00B0 | PCPDN         |
| 0x4000_0081 | <R0>          | 0x4000_0091 |               | 0x4000_00A1 |               | 0x4000_00B1 | <R0>          |
| 0x4000_0082 | <R0>          | 0x4000_0092 |               | 0x4000_00A2 |               | 0x4000_00B2 | <R0>          |
| 0x4000_0083 | <R0>          | 0x4000_0093 |               | 0x4000_00A3 |               | 0x4000_00B3 | <R0>          |
| 0x4000_0084 | PCCR          | 0x4000_0094 |               | 0x4000_00A4 |               | 0x4000_00B4 |               |
| 0x4000_0085 | <R0>          | 0x4000_0095 |               | 0x4000_00A5 |               | 0x4000_00B5 |               |
| 0x4000_0086 | <R0>          | 0x4000_0096 |               | 0x4000_00A6 |               | 0x4000_00B6 |               |
| 0x4000_0087 | <R0>          | 0x4000_0097 |               | 0x4000_00A7 |               | 0x4000_00B7 |               |
| 0x4000_0088 | PCFR1         | 0x4000_0098 |               | 0x4000_00A8 | PCOD          | 0x4000_00B8 | PCIE          |
| 0x4000_0089 | <R0>          | 0x4000_0099 |               | 0x4000_00A9 | <R0>          | 0x4000_00B9 | <R0>          |
| 0x4000_008A | <R0>          | 0x4000_009A |               | 0x4000_00AA | <R0>          | 0x4000_00BA | <R0>          |
| 0x4000_008B | <R0>          | 0x4000_009B |               | 0x4000_00AB | <R0>          | 0x4000_00BB | <R0>          |
| 0x4000_008C |               | 0x4000_009C |               | 0x4000_00AC | PCPUP         | 0x4000_00BC |               |
| 0x4000_008D |               | 0x4000_009D |               | 0x4000_00AD | <R0>          | 0x4000_00BD |               |
| 0x4000_008E |               | 0x4000_009E |               | 0x4000_00AE | <R0>          | 0x4000_00BE |               |
| 0x4000_008F |               | 0x4000_009F |               | 0x4000_00AF | <R0>          | 0x4000_00BF |               |

## [1] Port [2/5]

## &lt;PORT D&gt;

| 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 |               |

## &lt;PORT E&gt;

| 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 |               |

## &lt;PORT F&gt;

| 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 |               |

## [1] Port [3/5]

## &lt;PORT G&gt;

| 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 |               |

## &lt;PORT H&gt;

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4000_01C0 | PHDATA        | 0x4000_01D0 |               | 0x4000_01E0 |               | 0x4000_01F0 | PHPDN         |
| 0x4000_01C1 | <R0>          | 0x4000_01D1 |               | 0x4000_01E1 |               | 0x4000_01F1 | <R0>          |
| 0x4000_01C2 | <R0>          | 0x4000_01D2 |               | 0x4000_01E2 |               | 0x4000_01F2 | <R0>          |
| 0x4000_01C3 | <R0>          | 0x4000_01D3 |               | 0x4000_01E3 |               | 0x4000_01F3 | <R0>          |
| 0x4000_01C4 | PHCR          | 0x4000_01D4 |               | 0x4000_01E4 |               | 0x4000_01F4 |               |
| 0x4000_01C5 | <R0>          | 0x4000_01D5 |               | 0x4000_01E5 |               | 0x4000_01F5 |               |
| 0x4000_01C6 | <R0>          | 0x4000_01D6 |               | 0x4000_01E6 |               | 0x4000_01F6 |               |
| 0x4000_01C7 | <R0>          | 0x4000_01D7 |               | 0x4000_01E7 |               | 0x4000_01F7 |               |
| 0x4000_01C8 | PHFR1         | 0x4000_01D8 |               | 0x4000_01E8 | PHOD          | 0x4000_01F8 | PHIE          |
| 0x4000_01C9 | <R0>          | 0x4000_01D9 |               | 0x4000_01E9 | <R0>          | 0x4000_01F9 | <R0>          |
| 0x4000_01CA | <R0>          | 0x4000_01DA |               | 0x4000_01EA | <R0>          | 0x4000_01FA | <R0>          |
| 0x4000_01CB | <R0>          | 0x4000_01DB |               | 0x4000_01EB | <R0>          | 0x4000_01FB | <R0>          |
| 0x4000_01CC |               | 0x4000_01DC |               | 0x4000_01EC | PHPUP         | 0x4000_01FC |               |
| 0x4000_01CD |               | 0x4000_01DD |               | 0x4000_01ED | <R0>          | 0x4000_01FD |               |
| 0x4000_01CE |               | 0x4000_01DE |               | 0x4000_01EE | <R0>          | 0x4000_01FE |               |
| 0x4000_01CF |               | 0x4000_01DF |               | 0x4000_01EF | <R0>          | 0x4000_01FF |               |

## &lt;PORT I&gt;

| 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 |               |

[1] Port [4/5]

<PORT J>

| ADR         | Register name |
|-------------|---------------|
| 0x4000_0240 | PJDATA        |
| 0x4000_0241 | <R0>          |
| 0x4000_0242 | <R0>          |
| 0x4000_0243 | <R0>          |
| 0x4000_0244 | PJCR          |
| 0x4000_0245 | <R0>          |
| 0x4000_0246 | <R0>          |
| 0x4000_0247 | <R0>          |
| 0x4000_0248 | PJFR1         |
| 0x4000_0249 | <R0>          |
| 0x4000_024A | <R0>          |
| 0x4000_024B | <R0>          |
| 0x4000_024C |               |
| 0x4000_024D |               |
| 0x4000_024E |               |
| 0x4000_024F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_0250 |               |
| 0x4000_0251 |               |
| 0x4000_0252 |               |
| 0x4000_0253 |               |
| 0x4000_0254 |               |
| 0x4000_0255 |               |
| 0x4000_0256 |               |
| 0x4000_0257 |               |
| 0x4000_0258 |               |
| 0x4000_0259 |               |
| 0x4000_025A |               |
| 0x4000_025B |               |
| 0x4000_025C |               |
| 0x4000_025D |               |
| 0x4000_025E |               |
| 0x4000_025F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_0260 |               |
| 0x4000_0261 |               |
| 0x4000_0262 |               |
| 0x4000_0263 |               |
| 0x4000_0264 |               |
| 0x4000_0265 |               |
| 0x4000_0266 |               |
| 0x4000_0267 |               |
| 0x4000_0268 | PJOD          |
| 0x4000_0269 | <R0>          |
| 0x4000_026A | <R0>          |
| 0x4000_026B | <R0>          |
| 0x4000_026C | PJPUP         |
| 0x4000_026D | <R0>          |
| 0x4000_026E | <R0>          |
| 0x4000_026F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_0270 | PJPDN         |
| 0x4000_0271 | <R0>          |
| 0x4000_0272 | <R0>          |
| 0x4000_0273 | <R0>          |
| 0x4000_0274 |               |
| 0x4000_0275 |               |
| 0x4000_0276 |               |
| 0x4000_0277 |               |
| 0x4000_0278 | PJIE          |
| 0x4000_0279 | <R0>          |
| 0x4000_027A | <R0>          |
| 0x4000_027B | <R0>          |
| 0x4000_027C |               |
| 0x4000_027D |               |
| 0x4000_027E |               |
| 0x4000_027F |               |

<PORT K>

| ADR         | Register name |
|-------------|---------------|
| 0x4000_0280 | PKDATA        |
| 0x4000_0281 | <R0>          |
| 0x4000_0282 | <R0>          |
| 0x4000_0283 | <R0>          |
| 0x4000_0284 | PKCR          |
| 0x4000_0285 | <R0>          |
| 0x4000_0286 | <R0>          |
| 0x4000_0287 | <R0>          |
| 0x4000_0288 | PKFR1         |
| 0x4000_0289 | <R0>          |
| 0x4000_028A | <R0>          |
| 0x4000_028B | <R0>          |
| 0x4000_028C |               |
| 0x4000_028D |               |
| 0x4000_028E |               |
| 0x4000_028F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_0290 |               |
| 0x4000_0291 |               |
| 0x4000_0292 |               |
| 0x4000_0293 |               |
| 0x4000_0294 |               |
| 0x4000_0295 |               |
| 0x4000_0296 |               |
| 0x4000_0297 |               |
| 0x4000_0298 |               |
| 0x4000_0299 |               |
| 0x4000_029A |               |
| 0x4000_029B |               |
| 0x4000_029C |               |
| 0x4000_029D |               |
| 0x4000_029E |               |
| 0x4000_029F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_02A0 |               |
| 0x4000_02A1 |               |
| 0x4000_02A2 |               |
| 0x4000_02A3 |               |
| 0x4000_02A4 |               |
| 0x4000_02A5 |               |
| 0x4000_02A6 |               |
| 0x4000_02A7 |               |
| 0x4000_02A8 | PKOD          |
| 0x4000_02A9 | <R0>          |
| 0x4000_02AA | <R0>          |
| 0x4000_02AB | <R0>          |
| 0x4000_02AC | PKPUP         |
| 0x4000_02AD | <R0>          |
| 0x4000_02AE | <R0>          |
| 0x4000_02AF | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_02B0 | PKPDN         |
| 0x4000_02B1 | <R0>          |
| 0x4000_02B2 | <R0>          |
| 0x4000_02B3 | <R0>          |
| 0x4000_02B4 |               |
| 0x4000_02B5 |               |
| 0x4000_02B6 |               |
| 0x4000_02B7 |               |
| 0x4000_02B8 | PKIE          |
| 0x4000_02B9 | <R0>          |
| 0x4000_02BA | <R0>          |
| 0x4000_02BB | <R0>          |
| 0x4000_02BC |               |
| 0x4000_02BD |               |
| 0x4000_02BE |               |
| 0x4000_02BF |               |

<PORT L>

| ADR         | Register name |
|-------------|---------------|
| 0x4000_02C0 | PLDATA        |
| 0x4000_02C1 | <R0>          |
| 0x4000_02C2 | <R0>          |
| 0x4000_02C3 | <R0>          |
| 0x4000_02C4 |               |
| 0x4000_02C5 |               |
| 0x4000_02C6 |               |
| 0x4000_02C7 |               |
| 0x4000_02C8 | PLFR1         |
| 0x4000_02C9 | <R0>          |
| 0x4000_02CA | <R0>          |
| 0x4000_02CB | <R0>          |
| 0x4000_02CC |               |
| 0x4000_02CD |               |
| 0x4000_02CE |               |
| 0x4000_02CF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_02D0 |               |
| 0x4000_02D1 |               |
| 0x4000_02D2 |               |
| 0x4000_02D3 |               |
| 0x4000_02D4 |               |
| 0x4000_02D5 |               |
| 0x4000_02D6 |               |
| 0x4000_02D7 |               |
| 0x4000_02D8 |               |
| 0x4000_02D9 |               |
| 0x4000_02DA |               |
| 0x4000_02DB |               |
| 0x4000_02DC |               |
| 0x4000_02DD |               |
| 0x4000_02DE |               |
| 0x4000_02DF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_02E0 |               |
| 0x4000_02E1 |               |
| 0x4000_02E2 |               |
| 0x4000_02E3 |               |
| 0x4000_02E4 |               |
| 0x4000_02E5 |               |
| 0x4000_02E6 |               |
| 0x4000_02E7 |               |
| 0x4000_02E8 |               |
| 0x4000_02E9 |               |
| 0x4000_02EA |               |
| 0x4000_02EB |               |
| 0x4000_02EC |               |
| 0x4000_02ED |               |
| 0x4000_02EE |               |
| 0x4000_02EF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4000_02F0 |               |
| 0x4000_02F1 |               |
| 0x4000_02F2 |               |
| 0x4000_02F3 |               |
| 0x4000_02F4 |               |
| 0x4000_02F5 |               |
| 0x4000_02F6 |               |
| 0x4000_02F7 |               |
| 0x4000_02F8 | PLIE          |
| 0x4000_02F9 | <R0>          |
| 0x4000_02FA | <R0>          |
| 0x4000_02FB | <R0>          |
| 0x4000_02FC |               |
| 0x4000_02FD |               |
| 0x4000_02FE |               |
| 0x4000_02FF |               |

[1] Port [5/5]  
<PORT M>

| ADR         | レジスタ名  | ADR         | レジスタ名 | ADR         | レジスタ名 | ADR         | レジスタ名 |
|-------------|--------|-------------|-------|-------------|-------|-------------|-------|
| 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]

## &lt;TMRB0&gt;

| 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 |               |

## &lt;TMRB1&gt;

| 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 |               |

## &lt;TMRB2&gt;

| 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 |               |

## [2] 16-bit timer [2/3]

## &lt;TMRB3&gt;

| ADR         | Register name |
|-------------|---------------|
| 0x4001_00C0 | TB3EN         |
| 0x4001_00C1 | <R0>          |
| 0x4001_00C2 | <R0>          |
| 0x4001_00C3 | <R0>          |
| 0x4001_00C4 | TB3RUN        |
| 0x4001_00C5 | <R0>          |
| 0x4001_00C6 | <R0>          |
| 0x4001_00C7 | <R0>          |
| 0x4001_00C8 | TB3CR         |
| 0x4001_00C9 | <R0>          |
| 0x4001_00CA | <R0>          |
| 0x4001_00CB | <R0>          |
| 0x4001_00CC | TB3MOD        |
| 0x4001_00CD | <R0>          |
| 0x4001_00CE | <R0>          |
| 0x4001_00CF | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_00D0 | TB3FFCR       |
| 0x4001_00D1 | <R0>          |
| 0x4001_00D2 | <R0>          |
| 0x4001_00D3 | <R0>          |
| 0x4001_00D4 | TB3ST         |
| 0x4001_00D5 | <R0>          |
| 0x4001_00D6 | <R0>          |
| 0x4001_00D7 | <R0>          |
| 0x4001_00D8 | TB3IM         |
| 0x4001_00D9 | <R0>          |
| 0x4001_00DA | <R0>          |
| 0x4001_00DB | <R0>          |
| 0x4001_00DC | TB3UC         |
| 0x4001_00DD | "             |
| 0x4001_00DE | <R0>          |
| 0x4001_00DF | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_00E0 | TB3RG0        |
| 0x4001_00E1 | "             |
| 0x4001_00E2 | <R0>          |
| 0x4001_00E3 | <R0>          |
| 0x4001_00E4 | TB3RG1        |
| 0x4001_00E5 | "             |
| 0x4001_00E6 | <R0>          |
| 0x4001_00E7 | <R0>          |
| 0x4001_00E8 | TB3CP0        |
| 0x4001_00E9 | "             |
| 0x4001_00EA | <R0>          |
| 0x4001_00EB | <R0>          |
| 0x4001_00EC | TB3CP1        |
| 0x4001_00ED | "             |
| 0x4001_00EE | <R0>          |
| 0x4001_00EF | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_00F0 |               |
| 0x4001_00F1 |               |
| 0x4001_00F2 |               |
| 0x4001_00F3 |               |
| 0x4001_00F4 |               |
| 0x4001_00F5 |               |
| 0x4001_00F6 |               |
| 0x4001_00F7 |               |
| 0x4001_00F8 |               |
| 0x4001_00F9 |               |
| 0x4001_00FA |               |
| 0x4001_00FB |               |
| 0x4001_00FC |               |
| 0x4001_00FD |               |
| 0x4001_00FE |               |
| 0x4001_00FF |               |

## &lt;TMRB4&gt;

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0100 | TB4EN         |
| 0x4001_0101 | <R0>          |
| 0x4001_0102 | <R0>          |
| 0x4001_0103 | <R0>          |
| 0x4001_0104 | TB4RUN        |
| 0x4001_0105 | <R0>          |
| 0x4001_0106 | <R0>          |
| 0x4001_0107 | <R0>          |
| 0x4001_0108 | TB4CR         |
| 0x4001_0109 | <R0>          |
| 0x4001_010A | <R0>          |
| 0x4001_010B | <R0>          |
| 0x4001_010C | TB4MOD        |
| 0x4001_010D | <R0>          |
| 0x4001_010E | <R0>          |
| 0x4001_010F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0110 | TB4FFCR       |
| 0x4001_0111 | <R0>          |
| 0x4001_0112 | <R0>          |
| 0x4001_0113 | <R0>          |
| 0x4001_0114 | TB4ST         |
| 0x4001_0115 | <R0>          |
| 0x4001_0116 | <R0>          |
| 0x4001_0117 | <R0>          |
| 0x4001_0118 | TB4IM         |
| 0x4001_0119 | <R0>          |
| 0x4001_011A | <R0>          |
| 0x4001_011B | <R0>          |
| 0x4001_011C | TB4UC         |
| 0x4001_011D | "             |
| 0x4001_011E | <R0>          |
| 0x4001_011F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0120 | TB4RG0        |
| 0x4001_0121 | "             |
| 0x4001_0122 | <R0>          |
| 0x4001_0123 | <R0>          |
| 0x4001_0124 | TB4RG1        |
| 0x4001_0125 | "             |
| 0x4001_0126 | <R0>          |
| 0x4001_0127 | <R0>          |
| 0x4001_0128 | TB4CP0        |
| 0x4001_0129 | "             |
| 0x4001_012A | <R0>          |
| 0x4001_012B | <R0>          |
| 0x4001_012C | TB4CP1        |
| 0x4001_012D | "             |
| 0x4001_012E | <R0>          |
| 0x4001_012F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0130 |               |
| 0x4001_0131 |               |
| 0x4001_0132 |               |
| 0x4001_0133 |               |
| 0x4001_0134 |               |
| 0x4001_0135 |               |
| 0x4001_0136 |               |
| 0x4001_0137 |               |
| 0x4001_0138 |               |
| 0x4001_0139 |               |
| 0x4001_013A |               |
| 0x4001_013B |               |
| 0x4001_013C |               |
| 0x4001_013D | "             |
| 0x4001_013E |               |
| 0x4001_013F |               |

## &lt;TMRB5&gt;

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0140 | TB5EN         |
| 0x4001_0141 | <R0>          |
| 0x4001_0142 | <R0>          |
| 0x4001_0143 | <R0>          |
| 0x4001_0144 | TB5RUN        |
| 0x4001_0145 | <R0>          |
| 0x4001_0146 | <R0>          |
| 0x4001_0147 | <R0>          |
| 0x4001_0148 | TB5CR         |
| 0x4001_0149 | <R0>          |
| 0x4001_014A | <R0>          |
| 0x4001_014B | <R0>          |
| 0x4001_014C | TB5MOD        |
| 0x4001_014D | <R0>          |
| 0x4001_014E | <R0>          |
| 0x4001_014F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0150 | TB5FFCR       |
| 0x4001_0151 | <R0>          |
| 0x4001_0152 | <R0>          |
| 0x4001_0153 | <R0>          |
| 0x4001_0154 | TB5ST         |
| 0x4001_0155 | <R0>          |
| 0x4001_0156 | <R0>          |
| 0x4001_0157 | <R0>          |
| 0x4001_0158 | TB5IM         |
| 0x4001_0159 | <R0>          |
| 0x4001_015A | <R0>          |
| 0x4001_015B | <R0>          |
| 0x4001_015C | TB5UC         |
| 0x4001_015D | "             |
| 0x4001_015E | <R0>          |
| 0x4001_015F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0160 | TB5RG0        |
| 0x4001_0161 | "             |
| 0x4001_0162 | <R0>          |
| 0x4001_0163 | <R0>          |
| 0x4001_0164 | TB5RG1        |
| 0x4001_0165 | "             |
| 0x4001_0166 | <R0>          |
| 0x4001_0167 | <R0>          |
| 0x4001_0168 | TB5CP0        |
| 0x4001_0169 | "             |
| 0x4001_016A | <R0>          |
| 0x4001_016B | <R0>          |
| 0x4001_016C | TB5CP1        |
| 0x4001_016D | "             |
| 0x4001_016E | <R0>          |
| 0x4001_016F | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4001_0170 |               |
| 0x4001_0171 |               |
| 0x4001_0172 |               |
| 0x4001_0173 |               |
| 0x4001_0174 |               |
| 0x4001_0175 |               |
| 0x4001_0176 |               |
| 0x4001_0177 |               |
| 0x4001_0178 |               |
| 0x4001_0179 |               |
| 0x4001_017A |               |
| 0x4001_017B |               |
| 0x4001_017C |               |
| 0x4001_017D |               |
| 0x4001_017E |               |
| 0x4001_017F |               |

## [2] 16-bit timer [3/3]

## &lt;TMRB6&gt;

| 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 |               |

## &lt;TMRB7&gt;

| 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 |               |



22.1.3 [3] Encoder input (ENC)

<ENC0>

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4001_0400 | EN0TNCR       | 0x4001_0410 |               | 0x4001_0420 |               | 0x4001_0430 |               |
| 0x4001_0401 | "             | 0x4001_0411 |               | 0x4001_0421 |               | 0x4001_0431 |               |
| 0x4001_0402 | "             | 0x4001_0412 |               | 0x4001_0422 |               | 0x4001_0432 |               |
| 0x4001_0403 | <R0>          | 0x4001_0413 |               | 0x4001_0423 |               | 0x4001_0433 |               |
| 0x4001_0404 | EN0RELOAD     | 0x4001_0414 |               | 0x4001_0424 |               | 0x4001_0434 |               |
| 0x4001_0405 | "             | 0x4001_0415 |               | 0x4001_0425 |               | 0x4001_0435 |               |
| 0x4001_0406 | <R0>          | 0x4001_0416 |               | 0x4001_0426 |               | 0x4001_0436 |               |
| 0x4001_0407 | <R0>          | 0x4001_0417 |               | 0x4001_0427 |               | 0x4001_0437 |               |
| 0x4001_0408 | EN0INT        | 0x4001_0418 |               | 0x4001_0428 |               | 0x4001_0438 |               |
| 0x4001_0409 | "             | 0x4001_0419 |               | 0x4001_0429 |               | 0x4001_0439 |               |
| 0x4001_040A | "             | 0x4001_041A |               | 0x4001_042A |               | 0x4001_043A |               |
| 0x4001_040B | <R0>          | 0x4001_041B |               | 0x4001_042B |               | 0x4001_043B |               |
| 0x4001_040C | EN0CNT        | 0x4001_041C |               | 0x4001_042C |               | 0x4001_043C |               |
| 0x4001_040D | "             | 0x4001_041D |               | 0x4001_042D |               | 0x4001_043D |               |
| 0x4001_040E | "             | 0x4001_041E |               | 0x4001_042E |               | 0x4001_043E |               |
| 0x4001_040F | <R0>          | 0x4001_041F |               | 0x4001_042F |               | 0x4001_043F |               |

<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]

&lt;SIO0&gt;

| 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 |               |

&lt;SIO1&gt;

| 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 |               |

## [4] Serial interface (UART/SIO) [2/2]

## &lt;SIO2&gt;

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4002_0100 | SC2EN         | 0x4002_0110 | SC2BRCCR      | 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 |               |

## &lt;SIO3&gt;

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4002_0140 | SC3EN         | 0x4002_0150 | SC3BRCCR      | 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) [1/4]  
 <ADC A>

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4003_0000 | ADACLK        | 0x4003_0010 | ADACMPCR0     | 0x4003_0020 | ADAREG0       | 0x4003_0030 | ADAREG4       |
| 0x4003_0001 | "             | 0x4003_0011 | "             | 0x4003_0021 | "             | 0x4003_0031 | "             |
| 0x4003_0002 | "             | 0x4003_0012 | "             | 0x4003_0022 | "             | 0x4003_0032 | "             |
| 0x4003_0003 | "             | 0x4003_0013 | "             | 0x4003_0023 | "             | 0x4003_0033 | "             |
| 0x4003_0004 | ADAMOD0       | 0x4003_0014 | ADACMPCR1     | 0x4003_0024 | ADAREG1       | 0x4003_0034 | ADAREG5       |
| 0x4003_0005 | "             | 0x4003_0015 | "             | 0x4003_0025 | "             | 0x4003_0035 | "             |
| 0x4003_0006 | "             | 0x4003_0016 | "             | 0x4003_0026 | "             | 0x4003_0036 | "             |
| 0x4003_0007 | "             | 0x4003_0017 | "             | 0x4003_0027 | "             | 0x4003_0037 | "             |
| 0x4003_0008 | ADAMOD1       | 0x4003_0018 | ADACMP0       | 0x4003_0028 | ADAREG2       | 0x4003_0038 | ADAREG6       |
| 0x4003_0009 | "             | 0x4003_0019 | "             | 0x4003_0029 | "             | 0x4003_0039 | "             |
| 0x4003_000A | "             | 0x4003_001A | "             | 0x4003_002A | "             | 0x4003_003A | "             |
| 0x4003_000B | "             | 0x4003_001B | "             | 0x4003_002B | "             | 0x4003_003B | "             |
| 0x4003_000C | ADAMOD2       | 0x4003_001C | ADACMP1       | 0x4003_002C | ADAREG3       | 0x4003_003C | ADAREG7       |
| 0x4003_000D | "             | 0x4003_001D | "             | 0x4003_002D | "             | 0x4003_003D | "             |
| 0x4003_000E | "             | 0x4003_001E | "             | 0x4003_002E | "             | 0x4003_003E | "             |
| 0x4003_000F | "             | 0x4003_001F | "             | 0x4003_002F | "             | 0x4003_003F | "             |

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4003_0040 | ADAREG8       | 0x4003_0050 | ADAPSEL0      | 0x4003_0060 | ADAPSEL4      | 0x4003_0070 | ADAPSEL8      |
| 0x4003_0041 | "             | 0x4003_0051 | "             | 0x4003_0061 | "             | 0x4003_0071 | "             |
| 0x4003_0042 | "             | 0x4003_0052 | "             | 0x4003_0062 | "             | 0x4003_0072 | "             |
| 0x4003_0043 | "             | 0x4003_0053 | "             | 0x4003_0063 | "             | 0x4003_0073 | "             |
| 0x4003_0044 | ADAREG9       | 0x4003_0054 | ADAPSEL1      | 0x4003_0064 | ADAPSEL5      | 0x4003_0074 | ADAPSEL9      |
| 0x4003_0045 | "             | 0x4003_0055 | "             | 0x4003_0065 | "             | 0x4003_0075 | "             |
| 0x4003_0046 | "             | 0x4003_0056 | "             | 0x4003_0066 | "             | 0x4003_0076 | "             |
| 0x4003_0047 | "             | 0x4003_0057 | "             | 0x4003_0067 | "             | 0x4003_0077 | "             |
| 0x4003_0048 | ADAREG10      | 0x4003_0058 | ADAPSEL2      | 0x4003_0068 | ADAPSEL6      | 0x4003_0078 | ADAPSEL10     |
| 0x4003_0049 | "             | 0x4003_0059 | "             | 0x4003_0069 | "             | 0x4003_0079 | "             |
| 0x4003_004A | "             | 0x4003_005A | "             | 0x4003_006A | "             | 0x4003_007A | "             |
| 0x4003_004B | "             | 0x4003_005B | "             | 0x4003_006B | "             | 0x4003_007B | "             |
| 0x4003_004C | ADAREG11      | 0x4003_005C | ADAPSEL3      | 0x4003_006C | ADAPSEL7      | 0x4003_007C | ADAPSEL11     |
| 0x4003_004D | "             | 0x4003_005D | "             | 0x4003_006D | "             | 0x4003_007D | "             |
| 0x4003_004E | "             | 0x4003_005E | "             | 0x4003_006E | "             | 0x4003_007E | "             |
| 0x4003_004F | "             | 0x4003_005F | "             | 0x4003_006F | "             | 0x4003_007F | "             |

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4003_0080 | ADAPINTS0     | 0x4003_0090 | ADAPINTS4     | 0x4003_00A0 | ADAPSET2      | 0x4003_00B0 | ADATSET03     |
| 0x4003_0081 | "             | 0x4003_0091 | "             | 0x4003_00A1 | "             | 0x4003_00B1 | "             |
| 0x4003_0082 | "             | 0x4003_0092 | "             | 0x4003_00A2 | "             | 0x4003_00B2 | "             |
| 0x4003_0083 | "             | 0x4003_0093 | "             | 0x4003_00A3 | "             | 0x4003_00B3 | "             |
| 0x4003_0084 | ADAPINTS1     | 0x4003_0094 | ADAPINTS5     | 0x4003_00A4 | ADAPSET3      | 0x4003_00B4 | ADATSET47     |
| 0x4003_0085 | "             | 0x4003_0095 | "             | 0x4003_00A5 | "             | 0x4003_00B5 | "             |
| 0x4003_0086 | "             | 0x4003_0096 | "             | 0x4003_00A6 | "             | 0x4003_00B6 | "             |
| 0x4003_0087 | "             | 0x4003_0097 | "             | 0x4003_00A7 | "             | 0x4003_00B7 | "             |
| 0x4003_0088 | ADAPINTS2     | 0x4003_0098 | ADAPSET0      | 0x4003_00A8 | ADAPSET4      | 0x4003_00B8 | ADATSET811    |
| 0x4003_0089 | "             | 0x4003_0099 | "             | 0x4003_00A9 | "             | 0x4003_00B9 | "             |
| 0x4003_008A | "             | 0x4003_009A | "             | 0x4003_00AA | "             | 0x4003_00BA | "             |
| 0x4003_008B | "             | 0x4003_009B | "             | 0x4003_00AB | "             | 0x4003_00BB | "             |
| 0x4003_008C | ADAPINTS3     | 0x4003_009C | ADAPSET1      | 0x4003_00AC | ADAPSET5      | 0x4003_00BC | ADASSET03     |
| 0x4003_008D | "             | 0x4003_009D | "             | 0x4003_00AD | "             | 0x4003_00BD | "             |
| 0x4003_008E | "             | 0x4003_009E | "             | 0x4003_00AE | "             | 0x4003_00BE | "             |
| 0x4003_008F | "             | 0x4003_009F | "             | 0x4003_00AF | "             | 0x4003_00BF | "             |

[5] 12-bit A/D converter (A/DC) [2/4]

| ADR         | Register name |
|-------------|---------------|
| 0x4003_00C0 | ADASET47      |
| 0x4003_00C1 | "             |
| 0x4003_00C2 | "             |
| 0x4003_00C3 | "             |
| 0x4003_00C4 | ADASET811     |
| 0x4003_00C5 | "             |
| 0x4003_00C6 | "             |
| 0x4003_00C7 | "             |
| 0x4003_00C8 | ADASET03      |
| 0x4003_00C9 | "             |
| 0x4003_00CA | "             |
| 0x4003_00CB | "             |
| 0x4003_00CC | ADASET47      |
| 0x4003_00CD | "             |
| 0x4003_00CE | "             |
| 0x4003_00CF | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_00D0 | ADASET811     |
| 0x4003_00D1 | "             |
| 0x4003_00D2 | "             |
| 0x4003_00D3 | "             |
| 0x4003_00D4 | ADAMOD3       |
| 0x4003_00D5 | "             |
| 0x4003_00D6 | "             |
| 0x4003_00D7 | "             |
| 0x4003_00D8 |               |
| 0x4003_00D9 |               |
| 0x4003_00DA |               |
| 0x4003_00DB |               |
| 0x4003_00DC |               |
| 0x4003_00DD |               |
| 0x4003_00DE |               |
| 0x4003_00DF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_00E0 |               |
| 0x4003_00E1 |               |
| 0x4003_00E2 |               |
| 0x4003_00E3 |               |
| 0x4003_00E4 |               |
| 0x4003_00E5 |               |
| 0x4003_00E6 |               |
| 0x4003_00E7 |               |
| 0x4003_00E8 |               |
| 0x4003_00E9 |               |
| 0x4003_00EA |               |
| 0x4003_00EB |               |
| 0x4003_00EC |               |
| 0x4003_00ED |               |
| 0x4003_00EE |               |
| 0x4003_00EF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_00F0 |               |
| 0x4003_00F1 |               |
| 0x4003_00F2 |               |
| 0x4003_00F3 |               |
| 0x4003_00F4 |               |
| 0x4003_00F5 |               |
| 0x4003_00F6 |               |
| 0x4003_00F7 |               |
| 0x4003_00F8 |               |
| 0x4003_00F9 |               |
| 0x4003_00FA |               |
| 0x4003_00FB |               |
| 0x4003_00FC |               |
| 0x4003_00FD |               |
| 0x4003_00FE |               |
| 0x4003_00FF |               |

[5]12-bit A/D converter (A/DC) [3/4]

<ADC B>

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0200 | ADBCLK        |
| 0x4003_0201 | "             |
| 0x4003_0202 | "             |
| 0x4003_0203 | "             |
| 0x4003_0204 | ADBMOD0       |
| 0x4003_0205 | "             |
| 0x4003_0206 | "             |
| 0x4003_0207 | "             |
| 0x4003_0208 | ADBMOD1       |
| 0x4003_0209 | "             |
| 0x4003_020A | "             |
| 0x4003_020B | "             |
| 0x4003_020C | ADBMOD2       |
| 0x4003_020D | "             |
| 0x4003_020E | "             |
| 0x4003_020F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0210 | ADBCMPCR0     |
| 0x4003_0211 | "             |
| 0x4003_0212 | "             |
| 0x4003_0213 | "             |
| 0x4003_0214 | ADBCMPCR1     |
| 0x4003_0215 | "             |
| 0x4003_0216 | "             |
| 0x4003_0217 | "             |
| 0x4003_0218 | ADBCMP0       |
| 0x4003_0219 | "             |
| 0x4003_021A | "             |
| 0x4003_021B | "             |
| 0x4003_021C | ADBCMP1       |
| 0x4003_021D | "             |
| 0x4003_021E | "             |
| 0x4003_021F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0220 | ADBREG0       |
| 0x4003_0221 | "             |
| 0x4003_0222 | "             |
| 0x4003_0223 | "             |
| 0x4003_0224 | ADBREG1       |
| 0x4003_0225 | "             |
| 0x4003_0226 | "             |
| 0x4003_0227 | "             |
| 0x4003_0228 | ADBREG2       |
| 0x4003_0229 | "             |
| 0x4003_022A | "             |
| 0x4003_022B | "             |
| 0x4003_022C | ADBREG3       |
| 0x4003_022D | "             |
| 0x4003_022E | "             |
| 0x4003_022F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0230 | ADBREG4       |
| 0x4003_0231 | "             |
| 0x4003_0232 | "             |
| 0x4003_0233 | "             |
| 0x4003_0234 | ADBREG5       |
| 0x4003_0235 | "             |
| 0x4003_0236 | "             |
| 0x4003_0237 | "             |
| 0x4003_0238 | ADBREG6       |
| 0x4003_0239 | "             |
| 0x4003_023A | "             |
| 0x4003_023B | "             |
| 0x4003_023C | ADBREG7       |
| 0x4003_023D | "             |
| 0x4003_023E | "             |
| 0x4003_023F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0240 | ADBREG8       |
| 0x4003_0241 | "             |
| 0x4003_0242 | "             |
| 0x4003_0243 | "             |
| 0x4003_0244 | ADBREG9       |
| 0x4003_0245 | "             |
| 0x4003_0246 | "             |
| 0x4003_0247 | "             |
| 0x4003_0248 | ADBREG10      |
| 0x4003_0249 | "             |
| 0x4003_024A | "             |
| 0x4003_024B | "             |
| 0x4003_024C | ADBREG11      |
| 0x4003_024D | "             |
| 0x4003_024E | "             |
| 0x4003_024F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0250 | ADBPSEL0      |
| 0x4003_0251 | "             |
| 0x4003_0252 | "             |
| 0x4003_0253 | "             |
| 0x4003_0254 | ADBPSEL1      |
| 0x4003_0255 | "             |
| 0x4003_0256 | "             |
| 0x4003_0257 | "             |
| 0x4003_0258 | ADBPSEL2      |
| 0x4003_0259 | "             |
| 0x4003_025A | "             |
| 0x4003_025B | "             |
| 0x4003_025C | ADBPSEL3      |
| 0x4003_025D | "             |
| 0x4003_025E | "             |
| 0x4003_025F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0260 | ADBPSEL4      |
| 0x4003_0261 | "             |
| 0x4003_0262 | "             |
| 0x4003_0263 | "             |
| 0x4003_0264 | ADBPSEL5      |
| 0x4003_0265 | "             |
| 0x4003_0266 | "             |
| 0x4003_0267 | "             |
| 0x4003_0268 | ADBPSEL6      |
| 0x4003_0269 | "             |
| 0x4003_026A | "             |
| 0x4003_026B | "             |
| 0x4003_026C | ADBPSEL7      |
| 0x4003_026D | "             |
| 0x4003_026E | "             |
| 0x4003_026F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0270 | ADBPSEL8      |
| 0x4003_0271 | "             |
| 0x4003_0272 | "             |
| 0x4003_0273 | "             |
| 0x4003_0274 | ADBPSEL9      |
| 0x4003_0275 | "             |
| 0x4003_0276 | "             |
| 0x4003_0277 | "             |
| 0x4003_0278 | ADBPSEL10     |
| 0x4003_0279 | "             |
| 0x4003_027A | "             |
| 0x4003_027B | "             |
| 0x4003_027C | ADBPSEL11     |
| 0x4003_027D | "             |
| 0x4003_027E | "             |
| 0x4003_027F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0280 | ADBPINTS0     |
| 0x4003_0281 | "             |
| 0x4003_0282 | "             |
| 0x4003_0283 | "             |
| 0x4003_0284 | ADBPINTS1     |
| 0x4003_0285 | "             |
| 0x4003_0286 | "             |
| 0x4003_0287 | "             |
| 0x4003_0288 | ADBPINTS2     |
| 0x4003_0289 | "             |
| 0x4003_028A | "             |
| 0x4003_028B | "             |
| 0x4003_028C | ADBPINTS3     |
| 0x4003_028D | "             |
| 0x4003_028E | "             |
| 0x4003_028F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0290 | ADBPINTS4     |
| 0x4003_0291 | "             |
| 0x4003_0292 | "             |
| 0x4003_0293 | "             |
| 0x4003_0294 | ADBPINTS5     |
| 0x4003_0295 | "             |
| 0x4003_0296 | "             |
| 0x4003_0297 | "             |
| 0x4003_0298 | ADBPSET0      |
| 0x4003_0299 | "             |
| 0x4003_029A | "             |
| 0x4003_029B | "             |
| 0x4003_029C | ADBPSET1      |
| 0x4003_029D | "             |
| 0x4003_029E | "             |
| 0x4003_029F | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_02A0 | ADBPSET2      |
| 0x4003_02A1 | "             |
| 0x4003_02A2 | "             |
| 0x4003_02A3 | "             |
| 0x4003_02A4 | ADBPSET3      |
| 0x4003_02A5 | "             |
| 0x4003_02A6 | "             |
| 0x4003_02A7 | "             |
| 0x4003_02A8 | ADBPSET4      |
| 0x4003_02A9 | "             |
| 0x4003_02AA | "             |
| 0x4003_02AB | "             |
| 0x4003_02AC | ADBPSET5      |
| 0x4003_02AD | "             |
| 0x4003_02AE | "             |
| 0x4003_02AF | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_02B0 | ADBTSET03     |
| 0x4003_02B1 | "             |
| 0x4003_02B2 | "             |
| 0x4003_02B3 | "             |
| 0x4003_02B4 | ADBTSET47     |
| 0x4003_02B5 | "             |
| 0x4003_02B6 | "             |
| 0x4003_02B7 | "             |
| 0x4003_02B8 | ADBTSET811    |
| 0x4003_02B9 | "             |
| 0x4003_02BA | "             |
| 0x4003_02BB | "             |
| 0x4003_02BC | ADBSSET03     |
| 0x4003_02BD | "             |
| 0x4003_02BE | "             |
| 0x4003_02BF | "             |

## [5] 12-bit A/D converter (A/DC) [4/4]

| ADR         | Register name |
|-------------|---------------|
| 0x4003_02C0 | ADBSSET47     |
| 0x4003_02C1 | "             |
| 0x4003_02C2 | "             |
| 0x4003_02C3 | "             |
| 0x4003_02C4 | ADBSSET811    |
| 0x4003_02C5 | "             |
| 0x4003_02C6 | "             |
| 0x4003_02C7 | "             |
| 0x4003_02C8 | ADBASET03     |
| 0x4003_02C9 | "             |
| 0x4003_02CA | "             |
| 0x4003_02CB | "             |
| 0x4003_02CC | ADBASET47     |
| 0x4003_02CD | "             |
| 0x4003_02CE | "             |
| 0x4003_02CF | "             |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_02D0 | ADBASET811    |
| 0x4003_02D1 | "             |
| 0x4003_02D2 | "             |
| 0x4003_02D3 | "             |
| 0x4003_02D4 | ADBMOD3       |
| 0x4003_02D5 | "             |
| 0x4003_02D6 | "             |
| 0x4003_02D7 | "             |
| 0x4003_02D8 |               |
| 0x4003_02D9 |               |
| 0x4003_02DA |               |
| 0x4003_02DB |               |
| 0x4003_02DC |               |
| 0x4003_02DD |               |
| 0x4003_02DE |               |
| 0x4003_02DF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_02E0 |               |
| 0x4003_02E1 |               |
| 0x4003_02E2 |               |
| 0x4003_02E3 |               |
| 0x4003_02E4 |               |
| 0x4003_02E5 |               |
| 0x4003_02E6 |               |
| 0x4003_02E7 |               |
| 0x4003_02E8 |               |
| 0x4003_02E9 |               |
| 0x4003_02EA |               |
| 0x4003_02EB |               |
| 0x4003_02EC |               |
| 0x4003_02ED |               |
| 0x4003_02EE |               |
| 0x4003_02EF |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_02F0 |               |
| 0x4003_02F1 |               |
| 0x4003_02F2 |               |
| 0x4003_02F3 |               |
| 0x4003_02F4 |               |
| 0x4003_02F5 |               |
| 0x4003_02F6 |               |
| 0x4003_02F7 |               |
| 0x4003_02F8 |               |
| 0x4003_02F9 |               |
| 0x4003_02FA |               |
| 0x4003_02FB |               |
| 0x4003_02FC |               |
| 0x4003_02FD |               |
| 0x4003_02FE |               |
| 0x4003_02FF |               |

## 22.1.6 [6] OP-amp / Comparator (AMP/CMP)

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0400 | AMPCTLA       |
| 0x4003_0401 |               |
| 0x4003_0402 |               |
| 0x4003_0403 |               |
| 0x4003_0404 | Reserved      |
| 0x4003_0405 |               |
| 0x4003_0406 |               |
| 0x4003_0407 |               |
| 0x4003_0408 | AMPCTLB       |
| 0x4003_0409 |               |
| 0x4003_040A |               |
| 0x4003_040B |               |
| 0x4003_040C | Reserved      |
| 0x4003_040D |               |
| 0x4003_040E |               |
| 0x4003_040F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0410 | AMPCTLC       |
| 0x4003_0411 |               |
| 0x4003_0412 |               |
| 0x4003_0413 |               |
| 0x4003_0414 | Reserved      |
| 0x4003_0415 |               |
| 0x4003_0416 |               |
| 0x4003_0417 |               |
| 0x4003_0418 | AMPCTLD       |
| 0x4003_0419 |               |
| 0x4003_041A |               |
| 0x4003_041B |               |
| 0x4003_041C | Reserved      |
| 0x4003_041D |               |
| 0x4003_041E |               |
| 0x4003_041F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0420 | CMPCTLA       |
| 0x4003_0421 |               |
| 0x4003_0422 |               |
| 0x4003_0423 |               |
| 0x4003_0424 | Reserved      |
| 0x4003_0425 |               |
| 0x4003_0426 |               |
| 0x4003_0427 |               |
| 0x4003_0428 | CMPCTLB       |
| 0x4003_0429 |               |
| 0x4003_042A |               |
| 0x4003_042B |               |
| 0x4003_042C | Reserved      |
| 0x4003_042D |               |
| 0x4003_042E |               |
| 0x4003_042F |               |

| ADR         | Register name |
|-------------|---------------|
| 0x4003_0430 | CMPCTLC       |
| 0x4003_0431 |               |
| 0x4003_0432 |               |
| 0x4003_0433 |               |
| 0x4003_0434 | Reserved      |
| 0x4003_0435 |               |
| 0x4003_0436 |               |
| 0x4003_0437 |               |
| 0x4003_0438 | CMPCTLD       |
| 0x4003_0439 |               |
| 0x4003_043A |               |
| 0x4003_043B |               |
| 0x4003_043C | Reserved      |
| 0x4003_043D |               |
| 0x4003_043E |               |
| 0x4003_043F |               |

## 22.1.7 [7] 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 |               |

## 22.1.8 [8] 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 | CGPLSEL       | 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 |               |



22.1.9 [9] 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 |               |

22.1.10 [10] 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 |               |

## 22.1.11 [11] 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 | VETPMMO       | 0x4005_0060 | VEIQREF0      | 0x4005_0070 | VECIDKPO      |
| 0x4005_0041 | "             | 0x4005_0051 | "             | 0x4005_0061 | "             | 0x4005_0071 | "             |
| 0x4005_0042 | "             | 0x4005_0052 | <R0>          | 0x4005_0062 | <R0>          | 0x4005_0072 | <R0>          |
| 0x4005_0043 | "             | 0x4005_0053 | <R0>          | 0x4005_0063 | <R0>          | 0x4005_0073 | <R0>          |
| 0x4005_0044 | VEMCTLFO      | 0x4005_0054 | VEOMEGA0      | 0x4005_0064 | VEVDO         | 0x4005_0074 | VECIQK10      |
| 0x4005_0045 | "             | 0x4005_0055 | "             | 0x4005_0065 | <R0>          | 0x4005_0075 | "             |
| 0x4005_0046 | <R0>          | 0x4005_0056 | <R0>          | 0x4005_0066 | <R0>          | 0x4005_0076 | <R0>          |
| 0x4005_0047 | <R0>          | 0x4005_0057 | <R0>          | 0x4005_0067 | <R0>          | 0x4005_0077 | <R0>          |
| 0x4005_0048 | VEMODE0       | 0x4005_0058 | VETHETA0      | 0x4005_0068 | VEVQ0         | 0x4005_0078 | VECIQKPO      |
| 0x4005_0049 | <R0>          | 0x4005_0059 | "             | 0x4005_0069 | <R0>          | 0x4005_0079 | "             |
| 0x4005_004A | <R0>          | 0x4005_005A | <R0>          | 0x4005_006A | <R0>          | 0x4005_007A | <R0>          |
| 0x4005_004B | <R0>          | 0x4005_005B | <R0>          | 0x4005_006B | <R0>          | 0x4005_007B | <R0>          |
| 0x4005_004C | VEFMODE0      | 0x4005_005C | VEIDREF0      | 0x4005_006C | VECIDK10      | 0x4005_007C | VEVDIH0       |
| 0x4005_004D | "             | 0x4005_005D | "             | 0x4005_006D | "             | 0x4005_007D | "             |
| 0x4005_004E | <R0>          | 0x4005_005E | <R0>          | 0x4005_006E | <R0>          | 0x4005_007E | <R0>          |
| 0x4005_004F | <R0>          | 0x4005_005F | <R0>          | 0x4005_006F | <R0>          | 0x4005_007F | <R0>          |

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4005_0080 | VEVDILH0      | 0x4005_0090 | VEVMDPRD0     | 0x4005_00A0 | VECOS0        | 0x4005_00B0 | VESECTOR0     |
| 0x4005_0081 | "             | 0x4005_0091 | "             | 0x4005_00A1 | "             | 0x4005_00B1 | <R0>          |
| 0x4005_0082 | <R0>          | 0x4005_0092 | <R0>          | 0x4005_00A2 | <R0>          | 0x4005_00B2 | <R0>          |
| 0x4005_0083 | <R0>          | 0x4005_0093 | <R0>          | 0x4005_00A3 | <R0>          | 0x4005_00B3 | <R0>          |
| 0x4005_0084 | VEVQIH0       | 0x4005_0094 | VEMINPLS0     | 0x4005_00A4 | VESIN0        | 0x4005_00B4 | VESECTORM0    |
| 0x4005_0085 | "             | 0x4005_0095 | "             | 0x4005_00A5 | "             | 0x4005_00B5 | <R0>          |
| 0x4005_0086 | <R0>          | 0x4005_0096 | <R0>          | 0x4005_00A6 | <R0>          | 0x4005_00B6 | <R0>          |
| 0x4005_0087 | <R0>          | 0x4005_0097 | <R0>          | 0x4005_00A7 | <R0>          | 0x4005_00B7 | <R0>          |
| 0x4005_0088 | VEVQILH0      | 0x4005_0098 | VETRGCR0      | 0x4005_00A8 | VECOSM0       | 0x4005_00B8 | VEIA00        |
| 0x4005_0089 | "             | 0x4005_0099 | "             | 0x4005_00A9 | "             | 0x4005_00B9 | "             |
| 0x4005_008A | <R0>          | 0x4005_009A | <R0>          | 0x4005_00AA | <R0>          | 0x4005_00BA | <R0>          |
| 0x4005_008B | <R0>          | 0x4005_009B | <R0>          | 0x4005_00AB | <R0>          | 0x4005_00BB | <R0>          |
| 0x4005_008C | VEFPMMCHG0    | 0x4005_009C | Reserved      | 0x4005_00AC | VESINM0       | 0x4005_00BC | VEIB00        |
| 0x4005_008D | "             | 0x4005_009D | Reserved      | 0x4005_00AD | "             | 0x4005_00BD | "             |
| 0x4005_008E | <R0>          | 0x4005_009E | Reserved      | 0x4005_00AE | <R0>          | 0x4005_00BE | <R0>          |
| 0x4005_008F | <R0>          | 0x4005_009F | Reserved      | 0x4005_00AF | <R0>          | 0x4005_00BF | <R0>          |

## [11] Vector engine (VE) [2/3]

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4005_00C0 | VEIC00        | 0x4005_00D0 | VEVDC0        | 0x4005_00E0 | VEMODE1       | 0x4005_00F0 | VETHETA1      |
| 0x4005_00C1 | "             | 0x4005_00D1 | "             | 0x4005_00E1 | <R0>          | 0x4005_00F1 | "             |
| 0x4005_00C2 | <R0>          | 0x4005_00D2 | <R0>          | 0x4005_00E2 | <R0>          | 0x4005_00F2 | <R0>          |
| 0x4005_00C3 | <R0>          | 0x4005_00D3 | <R0>          | 0x4005_00E3 | <R0>          | 0x4005_00F3 | <R0>          |
| 0x4005_00C4 | VEIAADC0      | 0x4005_00D4 | VEID0         | 0x4005_00E4 | VEFMODE1      | 0x4005_00F4 | VEIDREF1      |
| 0x4005_00C5 | "             | 0x4005_00D5 | "             | 0x4005_00E5 | "             | 0x4005_00F5 | "             |
| 0x4005_00C6 | <R0>          | 0x4005_00D6 | "             | 0x4005_00E6 | <R0>          | 0x4005_00F6 | <R0>          |
| 0x4005_00C7 | <R0>          | 0x4005_00D7 | "             | 0x4005_00E7 | <R0>          | 0x4005_00F7 | <R0>          |
| 0x4005_00C8 | VEIBADC0      | 0x4005_00D8 | VEIQ0         | 0x4005_00E8 | VETPWM1       | 0x4005_00F8 | VEIQREF1      |
| 0x4005_00C9 | "             | 0x4005_00D9 | "             | 0x4005_00E9 | "             | 0x4005_00F9 | "             |
| 0x4005_00CA | <R0>          | 0x4005_00DA | "             | 0x4005_00EA | <R0>          | 0x4005_00FA | <R0>          |
| 0x4005_00CB | <R0>          | 0x4005_00DB | "             | 0x4005_00EB | <R0>          | 0x4005_00FB | <R0>          |
| 0x4005_00CC | VEICADC0      | 0x4005_00DC | VECTLF1       | 0x4005_00EC | VEOMEGA1      | 0x4005_00FC | VEVD1         |
| 0x4005_00CD | "             | 0x4005_00DD | <R0>          | 0x4005_00ED | "             | 0x4005_00FD | <R0>          |
| 0x4005_00CE | <R0>          | 0x4005_00DE | <R0>          | 0x4005_00EE | <R0>          | 0x4005_00FE | <R0>          |
| 0x4005_00CF | <R0>          | 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 | VECIQK1       | 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 | VECIDK11      | 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 | VECIQK11      | 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 | VEIA01        | 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 | VEIB01        | 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 | VEIC01        | 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 | VEVCMPU0      |
| 0x4005_014D | <R0>          | 0x4005_015D | "             | 0x4005_016D | "             | 0x4005_017D | "             |
| 0x4005_014E | <R0>          | 0x4005_015E | <R0>          | 0x4005_016E | "             | 0x4005_017E | <R0>          |
| 0x4005_014F | <R0>          | 0x4005_015F | <R0>          | 0x4005_016F | "             | 0x4005_017F | <R0>          |

[11] Vector engine (VE) [3/3]

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4005_0180 | VEVCMPO0      | 0x4005_0190 | VEVTRGCMPO0   | 0x4005_01A0 | VEVCMPO1      | 0x4005_01B0 | VEVTRGCMPO1   |
| 0x4005_0181 | "             | 0x4005_0191 | "             | 0x4005_01A1 | "             | 0x4005_01B1 | "             |
| 0x4005_0182 | <R0>          | 0x4005_0192 | <R0>          | 0x4005_01A2 | <R0>          | 0x4005_01B2 | <R0>          |
| 0x4005_0183 | <R0>          | 0x4005_0193 | <R0>          | 0x4005_01A3 | <R0>          | 0x4005_01B3 | <R0>          |
| 0x4005_0184 | VEVCMPO1      | 0x4005_0194 | VEVTRGSELO    | 0x4005_01A4 | VEVCMPO1      | 0x4005_01B4 | VEVTRGSEL1    |
| 0x4005_0185 | "             | 0x4005_0195 | <R0>          | 0x4005_01A5 | "             | 0x4005_01B5 | <R0>          |
| 0x4005_0186 | <R0>          | 0x4005_0196 | <R0>          | 0x4005_01A6 | <R0>          | 0x4005_01B6 | <R0>          |
| 0x4005_0187 | <R0>          | 0x4005_0197 | <R0>          | 0x4005_01A7 | <R0>          | 0x4005_01B7 | <R0>          |
| 0x4005_0188 | VEOUTCR0      | 0x4005_0198 | VEEMGRSO      | 0x4005_01A8 | VEOUTCR1      | 0x4005_01B8 | VEEMGRS1      |
| 0x4005_0189 | "             | 0x4005_0199 | <R0>          | 0x4005_01A9 | "             | 0x4005_01B9 | <R0>          |
| 0x4005_018A | <R0>          | 0x4005_019A | <R0>          | 0x4005_01AA | <R0>          | 0x4005_01BA | <R0>          |
| 0x4005_018B | <R0>          | 0x4005_019B | <R0>          | 0x4005_01AB | <R0>          | 0x4005_01BB | <R0>          |
| 0x4005_018C | VEVTRGCMPO0   | 0x4005_019C | VEVCMPO1      | 0x4005_01AC | VEVTRGCMPO1   | 0x4005_01BC | Reserved      |
| 0x4005_018D | "             | 0x4005_019D | "             | 0x4005_01AD | "             | 0x4005_01BD | Reserved      |
| 0x4005_018E | <R0>          | 0x4005_019E | <R0>          | 0x4005_01AE | <R0>          | 0x4005_01BE | Reserved      |
| 0x4005_018F | <R0>          | 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 | Reserved      |

## 22.1.12 [12] Programmable motor driver (PMD) [1/2]

| ADR         | Register name | ADR         | Register name | ADR         | Register name | ADR         | Register name |
|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|
| 0x4005_0400 | PMD0MDEN      | 0x4005_0410 | PMD0MDCNT0    | 0x4005_0420 | PMD0CMPW      | 0x4005_0430 | PMD0EMGREL    |
| 0x4005_0401 | <R0>          | 0x4005_0411 | "             | 0x4005_0421 | "             | 0x4005_0431 | <R0>          |
| 0x4005_0402 | <R0>          | 0x4005_0412 | <R0>          | 0x4005_0422 | <R0>          | 0x4005_0432 | <R0>          |
| 0x4005_0403 | <R0>          | 0x4005_0413 | <R0>          | 0x4005_0423 | <R0>          | 0x4005_0433 | <R0>          |
| 0x4005_0404 | PMD0PORTMD    | 0x4005_0414 | PMD0MDPRD     | 0x4005_0424 | PMD0MODESEL   | 0x4005_0434 | PMD0EMGCR     |
| 0x4005_0405 | <R0>          | 0x4005_0415 | "             | 0x4005_0425 | <R0>          | 0x4005_0435 | "             |
| 0x4005_0406 | <R0>          | 0x4005_0416 | <R0>          | 0x4005_0426 | <R0>          | 0x4005_0436 | <R0>          |
| 0x4005_0407 | <R0>          | 0x4005_0417 | <R0>          | 0x4005_0427 | <R0>          | 0x4005_0437 | <R0>          |
| 0x4005_0408 | PMD0MDCR      | 0x4005_0418 | PMD0CMPU      | 0x4005_0428 | PMD0MDOU      | 0x4005_0438 | PMD0EMGSTA    |
| 0x4005_0409 | <R0>          | 0x4005_0419 | "             | 0x4005_0429 | "             | 0x4005_0439 | <R0>          |
| 0x4005_040A | <R0>          | 0x4005_041A | <R0>          | 0x4005_042A | <R0>          | 0x4005_043A | <R0>          |
| 0x4005_040B | <R0>          | 0x4005_041B | <R0>          | 0x4005_042B | <R0>          | 0x4005_043B | <R0>          |
| 0x4005_040C | PMD0CNTSTA    | 0x4005_041C | PMD0CMPV      | 0x4005_042C | PMDMDPOT      | 0x4005_043C | PMD0VOCR      |
| 0x4005_040D | <R0>          | 0x4005_041D | "             | 0x4005_042D | "             | 0x4005_043D | "             |
| 0x4005_040E | <R0>          | 0x4005_041E | "             | 0x4005_042E | <R0>          | 0x4005_043E | <R0>          |
| 0x4005_040F | <R0>          | 0x4005_041F | "             | 0x4005_042F | <R0>          | 0x4005_043F | <R0>          |
| 0x4005_0440 | PMD0OVVSTA    | 0x4005_0450 | PMD0TRGCM2    | 0x4005_0460 | PMD0TRGSEL    | 0x4005_0470 | Reserved      |
| 0x4005_0441 | <R0>          | 0x4005_0451 | "             | 0x4005_0461 | "             | 0x4005_0471 | Reserved      |
| 0x4005_0442 | <R0>          | 0x4005_0452 | <R0>          | 0x4005_0462 | "             | 0x4005_0472 | Reserved      |
| 0x4005_0443 | <R0>          | 0x4005_0453 | <R0>          | 0x4005_0463 | "             | 0x4005_0473 | Reserved      |
| 0x4005_0444 | PMD0DTR       | 0x4005_0454 | PMD0TRGCM3    | 0x4005_0464 | Reserved      | 0x4005_0474 | Reserved      |
| 0x4005_0445 | <R0>          | 0x4005_0455 | "             | 0x4005_0465 | Reserved      | 0x4005_0475 | Reserved      |
| 0x4005_0446 | <R0>          | 0x4005_0456 | <R0>          | 0x4005_0466 | Reserved      | 0x4005_0476 | Reserved      |
| 0x4005_0447 | <R0>          | 0x4005_0457 | <R0>          | 0x4005_0467 | Reserved      | 0x4005_0477 | Reserved      |
| 0x4005_0448 | PMD0TRGCM0    | 0x4005_0458 | PMD0TRGCR     | 0x4005_0468 | Reserved      | 0x4005_0478 | Reserved      |
| 0x4005_0449 | "             | 0x4005_0459 | "             | 0x4005_0469 | Reserved      | 0x4005_0479 | Reserved      |
| 0x4005_044A | <R0>          | 0x4005_045A | <R0>          | 0x4005_046A | Reserved      | 0x4005_047A | Reserved      |
| 0x4005_044B | <R0>          | 0x4005_045B | <R0>          | 0x4005_046B | Reserved      | 0x4005_047B | Reserved      |
| 0x4005_044C | PMD0TRGCM1    | 0x4005_045C | PMD0TRGMD     | 0x4005_046C | Reserved      | 0x4005_047C | Reserved      |
| 0x4005_044D | "             | 0x4005_045D | "             | 0x4005_046D | Reserved      | 0x4005_047D | Reserved      |
| 0x4005_044E | <R0>          | 0x4005_045E | "             | 0x4005_046E | Reserved      | 0x4005_047E | Reserved      |
| 0x4005_044F | <R0>          | 0x4005_045F | "             | 0x4005_046F | Reserved      | 0x4005_047F | Reserved      |

[12] Programmable motor driver (PMD) [2/2]

| 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 | PMD1TRGCMPO   |
| 0x4005_04C9 | "             |
| 0x4005_04CA | <R0>          |
| 0x4005_04CB | <R0>          |
| 0x4005_04CC | PMD1TRGCMPI   |
| 0x4005_04CD | "             |
| 0x4005_04CE | <R0>          |
| 0x4005_04CF | <R0>          |

| ADR         | Register name |
|-------------|---------------|
| 0x4005_04D0 | PMD1TRGCMPI2  |
| 0x4005_04D1 | "             |
| 0x4005_04D2 | <R0>          |
| 0x4005_04D3 | <R0>          |
| 0x4005_04D4 | PMD1TRGCMPI3  |
| 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      |

## 22.1.13 [13] 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 | <R0>          | 0x41FF_F031 | Reserved      |
| 0x41FF_F002 | Reserved      | 0x41FF_F012 | <R0>          | 0x41FF_F022 | Reserved      | 0x41FF_F032 | Reserved      |
| 0x41FF_F003 | Reserved      | 0x41FF_F013 | <R0>          | 0x41FF_F023 | Reserved      | 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      |

## 22.1.14 [14] 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      |

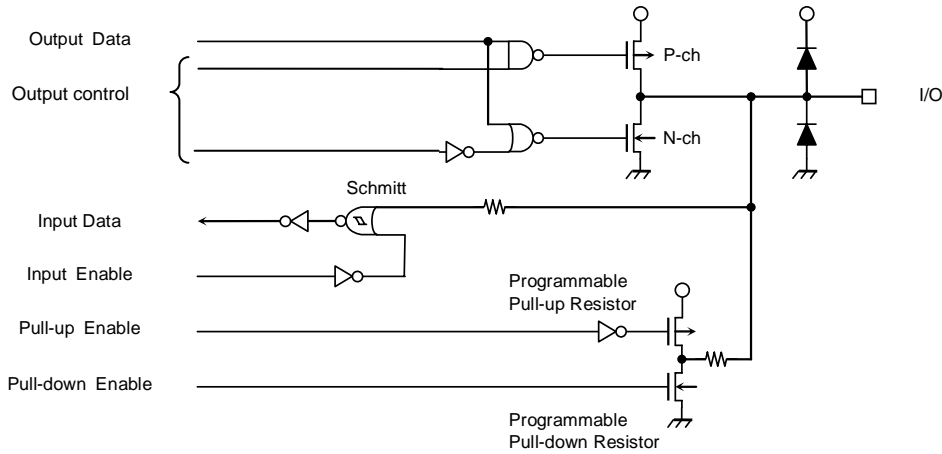
### 23. Port Section Equivalent Circuit Schematics

- How to read the schematics

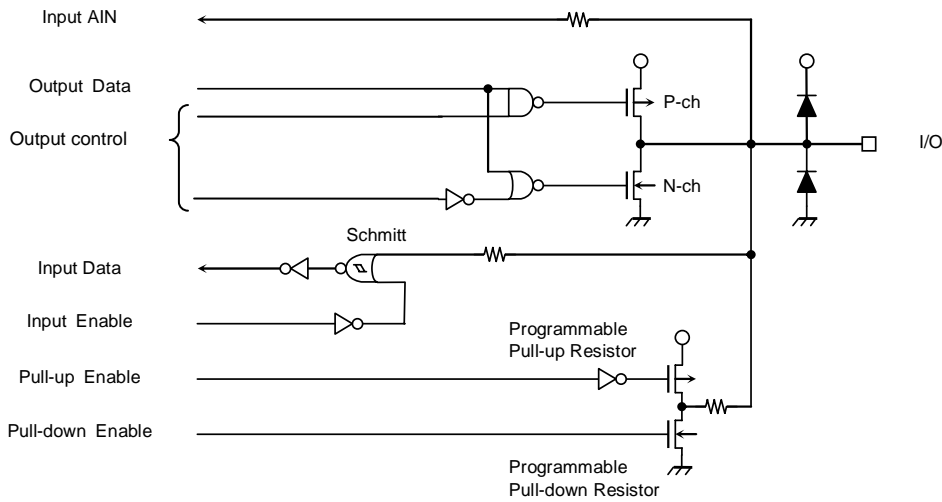
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

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 PC0-7, PD0-6, PE0-7, PF0-4, PG0-7

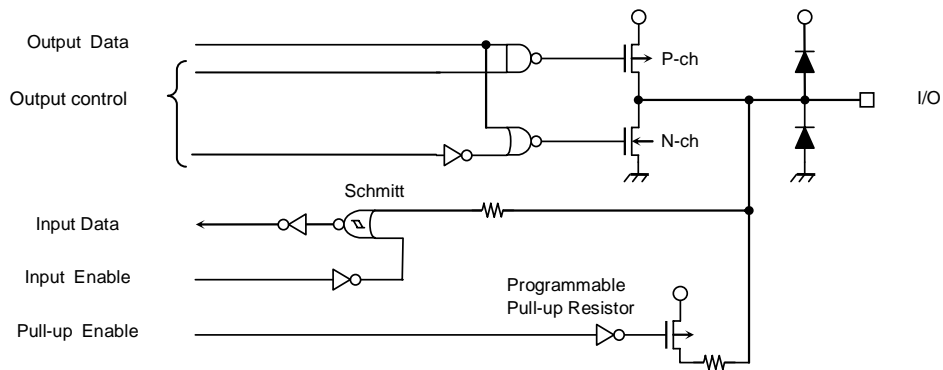


- PH0-7, PI0-3, PJ0-7, PK0-1

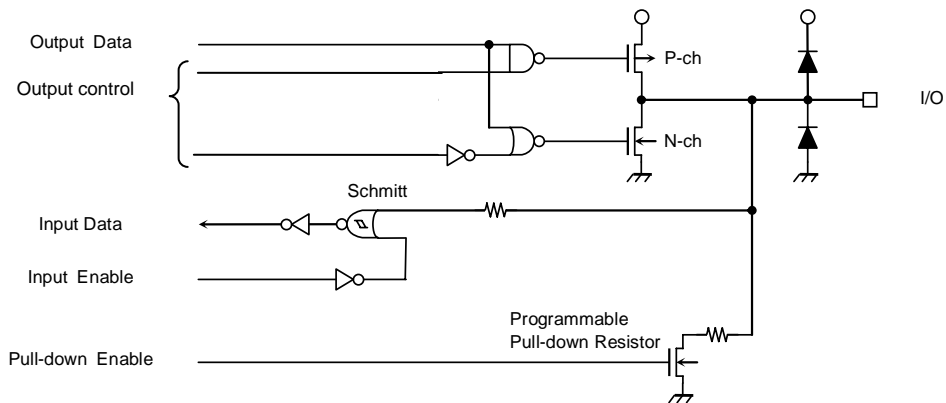




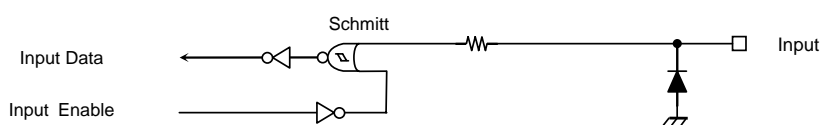
• PB0-3, PB5-7



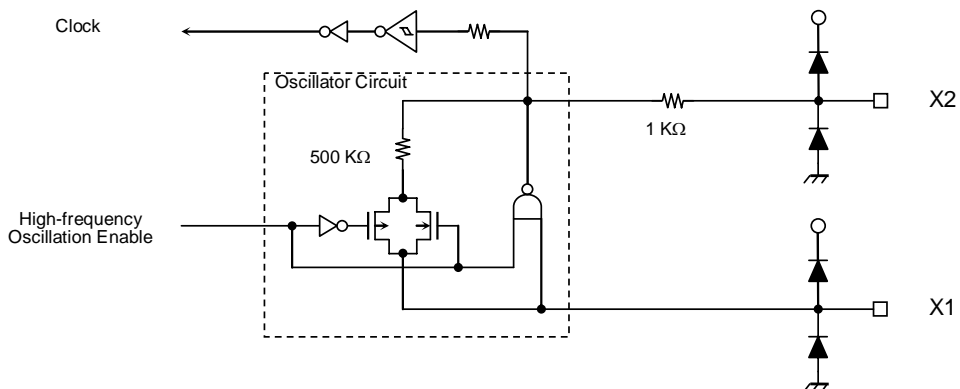
• PB4



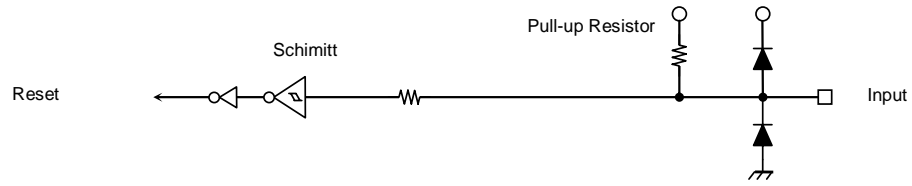
• PL0-1



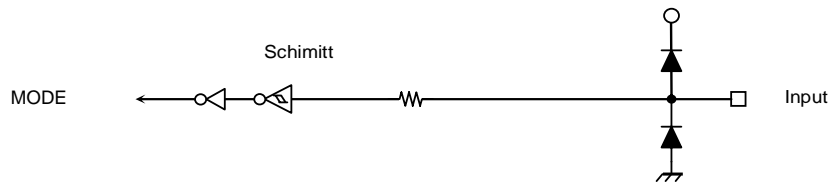
• X1, X2



- $\overline{\text{RESET}}$

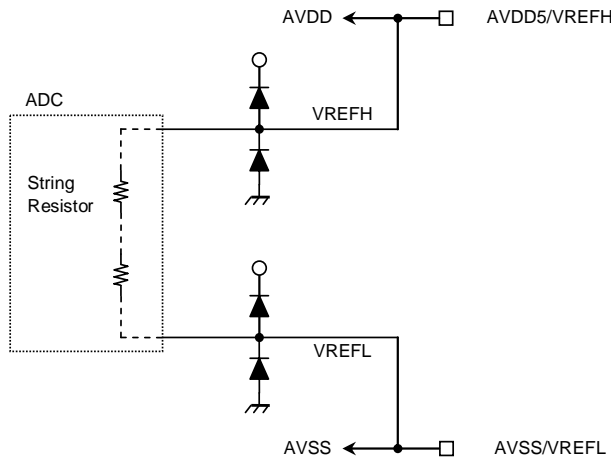


- MODE

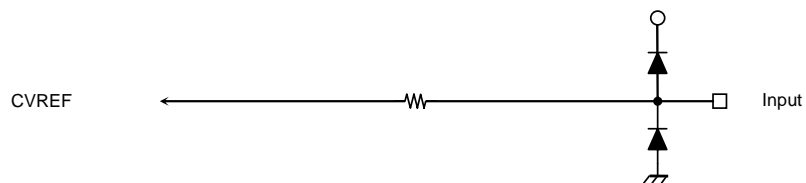


Note : MODE must be connected with GND.

- AVDD5A / VREFHA , AVSSA / VREFLA , AVDD5B / VREFHB , AVSSB / VREFLB



- CVREFABC, CVREFD



## 24 Electrical Characteristics

### 24.1 Absolute Maximum Ratings

| Parameter                     |         | Symbol              | Rating                  | Unit |
|-------------------------------|---------|---------------------|-------------------------|------|
| Supply voltage                |         | DVDD5               | - 0.3 to 6              | V    |
|                               |         | DVDD5E              | - 0.3 to 6              |      |
|                               |         | RVDD5               | - 0.3 to 6              |      |
|                               |         | AVDD5A/B            | - 0.3 to 6              |      |
|                               |         | AMPVDD5             | - 0.3 to 6              |      |
| Capacitor voltage             |         | VOUT15              | - 0.3 to 3              | V    |
|                               |         | VOUT3               | - 0.3 to 3.9            |      |
| Input voltage                 |         | VIN                 | - 0.3 to VDD+0.3(Note1) | V    |
| Low-level output current      | Per pin | I <sub>OL</sub>     | 5                       | mA   |
|                               | Total   | ΣI <sub>OL</sub>    | 50                      |      |
| High-level output current     | Per pin | I <sub>OH</sub>     | - 5                     |      |
|                               | Total   | ΣI <sub>OH</sub>    | - 50                    |      |
| Power consumption (Ta = 85°C) |         | PD                  | 600                     | mW   |
| Soldering temperature (10s)   |         | T <sub>SOLDER</sub> | 260                     | °C   |
| Storage temperature           |         | T <sub>STG</sub>    | - 55 to 125             | °C   |
| Operating Temperature         |         | T <sub>OPR</sub>    | - 40 to 85              | °C   |

**(Note)** Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

(Note 1) VDD = DVDD5 = DVDD5E = AVDD5A = AVDD5B = RVDD5 = AMPVDD5

## 24.2 DC Electrical Characteristics (1/2)

DVSS = AVSSA/B = AMPVSS = 0V, Ta = -40 to 85°C

| Parameter                                       |               | Symbol  | Rating  | Min.     | Typ.<br>(Note 1) | Max.     | Unit |
|---|---------------|---|---|----------|------------------|----------|------|
| Supply voltage<br>(Note2)                       |               | DVDD5<br>DVDD5E<br>AVDD5A<br>AVDD5B<br>RVDD5<br>AMPVDD5 | fosc = 8 to 10MHz<br>fsys = 1 to 80MHz                            | 4.5      | -                | 5.5      | V    |
| Supply voltage<br>(during Flash W/E)<br>(Note2) |               | DVDD5<br>DVDD5E<br>AVDD5A<br>AVDD5B<br>RVDD5<br>AMPVDD5 | fosc = 8 to 10MHz<br>fsys = 1 to 80MHz<br>( Ta ( °C ) = 0 to 70 ) | 4.5      | -                | 5.5      | V    |
| Low-level<br>input voltage                      | Schmitt-Input | $V_{IL1}$   | VDD = 4.5 to 5.5V   | -0.3     |                  | 0.25 VDD | V    |
|   | Schmitt-Input | $V_{IH1}$   | VDD = 4.5V to 5.5V  | 0.75 VDD |                  | VDD      |      |
| Capacitance for VOUT15 and<br>VOUT3 (Note 3)    |               | Cout  | RVDD5 = 4.5V to 5.5V<br>VOUT15,VOUT3                              | 3.3      |                  | 4.7      | μF   |
| Low-level output voltage                        |               | $V_{OL}$  | $I_{OL} = 1.6mA$<br>VDD 4.5V<br>(Note4)                           | -        | -                | 0.4      | V    |
| High-level output voltage                       |               | $V_{OH}$  | $I_{OH} = -1.6mA$<br>VDD 4.5V<br>(Note4)                          | 4.1      | -                | -        | V    |
| Input leakage<br>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<br>(Note4)                                 | -        | 0.05             | ± 10     |      |
| Pull-up resistor at Reset                       |               | $R_{RST}$   | 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    |
| Programmable pull-up/<br>pull-down resistor     |               | $P_{KH}$  | 4.5V VDD 5.5V (Note4)   | -        | 50               | 150      | kΩ   |
| Pin capacitance<br>(Except power supply pins)   |               | $C_{IO}$  | fc = 1MHz   | -        | -                | 10       | pF   |

(Note 1) Ta = 25°C, DVDD5 = DVDD5E = AVDD5A = AVDD5B = RVDD5 = AMPVDD5 = 5V, unless otherwise noted.

(Note 2) The same voltage must be supplied to DVDD5, DVDD5E, AVDD5A/B, RVDD5 and AMPVDD5.

(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 = DVDD5E = DVDD5 = AVDD5A = AVDD5B = AMPVDD5

### 24.3 DC Electrical Characteristics (2/2)

$T_a = -40$  to  $85^\circ\text{C}$  DVDD5 = DVDD5E = RVDD5 = AVDD5A/B = AMPVDD5 = 4.5V to 5.5V

| Parameter                | Symbol   | Rating  | Min. | Typ.<br>(Note 1) | Max. | Unit |
|--------------------------|----------|---|------|------------------|------|------|
| NORMAL (Note 2) Gear 1/1 | $I_{CC}$ | $f_{\text{sys}}=80\text{MHz}$ ( $f_{\text{osc}}=10\text{MHz}$ ) |      | 70               | 80   | mA   |
| IDLE (Note 2) Gear 1/1   |          |   |      | 21               | 30   |      |
| STOP                     |          |   |      | 7                | 11   | mA   |

(Note 1)  $T_a = 25^\circ\text{C}$ , DVDD5 = DVDD5E = AVDD5A/B = RVDD5 = AMPVDD5 = 5V, unless otherwise noted.

(Note 2)  $I_{CC}$  NORMAL, IDLE: All functions operates excluding A/D, Op amp and Comparator.

(Note 3) A/D reference voltage supply can not go into off state.

### 24.4 12-bit ADC Electrical Characteristics

DVDD5=RVDD5=AVDD5A/VREFHA=AVDD5B/VREFHB = 4.5V to 5.5V,  
DVSS=AVSSA/VREFLA=AVSSB/VREFLB=0V,  $T_a = -40$  to  $85^\circ\text{C}$

| Parameter                                  | Symbol            | Rating  | Min.                | Typ. | Max.      | Unit |
|--|-------------------|---|---------------------|------|-----------|------|
| Analog reference voltage (+)               | VREFHA<br>VREFHB  | –   |                     | AVDD |           | V    |
| Analog input voltage                       | VAIN              | –   | AVSS                | –    | AVDD      | V    |
| Analog supply current<br>(Note 1) (Note 2) | IREF              | DVSS = AVSS   | –                   | 3.5  | 4.5       | mA   |
| Supply current<br>(Note 1)                 | A/D<br>conversion | –   | Except IREF (Note2) | –    | 3         | mA   |
| INL error                                  | –                 | AIN resistance $\leq 600\Omega$<br>AIN load<br>capacitance $0.1\mu\text{F}$<br>Conversion time $2\mu\text{s}$ |                     |      | $\pm 6$   | LSB  |
| DNL error                                  |                   |   |                     |      | $\pm 5$   |      |
| Offset error                               |                   |   |                     |      | $\pm 5$   |      |
| Full-scale error                           |                   |   |                     |      | $\pm 5$   |      |
| Total error                                |                   |   |                     |      | +5 to -10 |      |

(Note 1) Current for one unit of ADC.

(Note 2) A/D reference voltage supply can not go into off state.

(Note 3)  $1\text{LSB} = (\text{AVDD} - \text{AVSS}) / 4096[\text{V}]$

(Note 4) AVDD = AVDD5A = AVDD5B, AVSS = AVSSA = AVSSB

(Note 5) The characteristic is measured under the condition in which the only ADC is operating.

## 24.5 Op-Amps Electrical Characteristics

DVDD5 = RVDD5 = AVDD5A/VREFHA = AVDD5B/VREFHB = AMPVDD5 = 4.5V to 5.5V,  
DVSS = AVSSA/VREFLA = AVSSB/VREFLB = AMPVSS = 0V, Ta = -40 to 85°C

| Parameter           | Symbol | Rating   | Min.                              | Typ.        | Max.                              | Unit |             |    |       |             |         |             |   |             |    |
|---------------------|--------|--|-----------------------------------|-------------|-----------------------------------|------|-------------|----|-------|-------------|---------|-------------|---|-------------|----|
| Gain (Note 1)       | VGAIN  | AVDD = 4.5 to 5.5V<br>AVSS = 0V  | × 1.5                             | –           | × 10                              |      |             |    |       |             |         |             |   |             |    |
| Input voltage range | VAMPIN |  | $\frac{(AVDD \times 0.1)}{VGAIN}$ | –           | $\frac{(AVDD \times 0.9)}{VGAIN}$ | V    |             |    |       |             |         |             |   |             |    |
| Offset voltage      | VOFF1  | <table border="1"> <tr> <td>VGAIN</td> <td>3.5</td> <td rowspan="2">– 6 × VGAIN</td> <td rowspan="2">–</td> <td rowspan="2">+ 6 × VGAIN</td> <td rowspan="2">mV</td> </tr> <tr> <td>VGAIN</td> <td>3</td> <td>Ta &lt; 70</td> </tr> </table> | VGAIN                             | 3.5         | – 6 × VGAIN                       | –    | + 6 × VGAIN | mV | VGAIN | 3           | Ta < 70 | – 6 × VGAIN | – | + 6 × VGAIN | mV |
|                     | VGAIN  |  | 3.5                               | – 6 × VGAIN |                                   |      |             |    | –     | + 6 × VGAIN | mV      |             |   |             |    |
| VGAIN               | 3      | Ta < 70  |                                   |             |                                   |      |             |    |       |             |         |             |   |             |    |
|                     | VOFF2  | <table border="1"> <tr> <td>Ta</td> <td>70</td> <td>–20</td> <td>–</td> <td>+20</td> <td></td> </tr> </table>  | Ta                                | 70          | –20                               | –    | +20         |    | –20   | –           | +20     |             |   |             |    |
| Ta                  | 70     | –20  | –                                 | +20         |                                   |      |             |    |       |             |         |             |   |             |    |
| Gain error          |        |  | –                                 | ± 1         | ± 3                               | %    |             |    |       |             |         |             |   |             |    |
| Slew rate (Note 2)  | Vthr   | AVDD = 4.5 to 5.5V<br>AVSS = 0V<br>5pF, VGAIN = × 2.5  | 2                                 | –           | –                                 | V/μs |             |    |       |             |         |             |   |             |    |

(Note 1) Gain can be selected among ×2.5, ×3, ×3.5, ×4, ×6 and ×8 by register setting.

(Note 2) Slew rate means a slant til the output of amplifier reaches AVDD-0.001×AVDD.

(Note 3) AVDD = AVDD5A = AVDD5B = AMPVDD5, AVSS = AVSSA = AVSSB = AMPVSS

## 24.6 Comparator Electrical Characteristics

DVDD5 = RVDD5 = AVDD5A/VREFHA = AVDD5B/VREFHB = AMPVDD5 = 4.5V to 5.5V,  
DVSS = AVSSA/VREFLA = AVSSB/VREFLB = 0V, Ta = -40 to 85°C

| Parameter                      | Symbol | Rating                          | Min. | Typ. | Max.     | Unit |
|--------------------------------|--------|---------------------------------|------|------|----------|------|
| Offset voltage                 | VOFF   | AVDD = 4.5 to 5.5V<br>AVSS = 0V | –    | ± 4  | –        | %    |
| AIN input voltage range        | VIN    |                                 | AVSS | –    | AVDD     | V    |
| Reference voltage range        | VREF   |                                 | 0.9  | –    | AVDD–0.2 | V    |
| Response time (Note 1) (Note2) |        |                                 | –    | –    | 1        | μs   |

(Note 1) 1.0V VREF AVDD-0.2V

(Note 2) The case that VIN varies from VREF-100mv to VREF+100mv or from VREF+100mv to VREF-100mv.

(Note 3) AVDD = AVDD5A = AVDD5B = AMPVDD5, AVSS = AVSSA = AVSSB = AMPVSS

## 24.7 AC Electrical Characteristics

AC measurement 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 = DVDD5E = DVDD5 = AVDD5A = AVDD5B = AMPVDD5

### 24.7.1 Serial Channel Timing (SIO)

(1) I/O Interface mode

SCLK input mode

(VDD = 4.5V to 5.5V, Ta = -40 to 85°C)

| Parameter  | Symbol           | Equation                            |      | 80MHz      |      | Unit |
|--|------------------|-------------------------------------|------|------------|------|------|
|  |                  | Min.                                | Max. | Min.       | Max. |      |
| SCLK Clock High width (input)                      | t <sub>SCH</sub> | 3x                                  |      | 37.5       |      | ns   |
| SCLK Clock Low width (input)                       | t <sub>SCL</sub> | 3x                                  |      | 37.5       |      | ns   |
| SCLK cycle   | t <sub>SCY</sub> | t <sub>SCH</sub> + t <sub>SCL</sub> |      | 75         |      | ns   |
| OutputData to SCLK rise or fall (Note 1)           | t <sub>OSS</sub> | t <sub>SCY</sub> /2 - 3x - 45       |      | 0 (Note 2) |      | ns   |
| OutputData hold or fall after SCLK rising (Note 1) | t <sub>OHS</sub> | t <sub>SCY</sub> /2                 |      | 37.5       |      | ns   |
| Input Data valid SCLK rise or fall (Note 1)        | t <sub>SRD</sub> | 30                                  |      | 30         |      | ns   |
| InputData hold or fall after SCLK rising (Note 1)  | t <sub>HSR</sub> | x + 30                              |      | 42.5       |      | ns   |

**(Note 1) SCLK rise or fall: Measured relative to the programmed active edge of SCLK.**

**(Note 2) t<sub>OSS</sub> should be always positive. Therefore, set proper SCLK parameters (t<sub>SCY</sub>, t<sub>SCL</sub> and t<sub>SCY</sub>) to keep t<sub>OSS</sub> positive. (t<sub>OSS</sub> > 0)**

SCLK output mode

(VDD = 4.5V to 5.5V, Ta = -40 to 85°C)

| Parameter                         | Symbol           | Equation                 |      | 80MHz |      | Unit |
|-----------------------------------|------------------|--------------------------|------|-------|------|------|
|                                   |                  | Min.                     | Max. | Min.  | Max. |      |
| SCLK cycle (programmable)         | t <sub>SCY</sub> | 4x                       |      | 50    |      | ns   |
| OutputData to SCLK rise           | t <sub>OSS</sub> | t <sub>SCY</sub> /2 - 20 |      | 5     |      | ns   |
| OutputData hold after SCLK rising | t <sub>OHS</sub> | t <sub>SCY</sub> /2 - 20 |      | 5     |      | ns   |
| InputData valid to SCLK rise      | t <sub>SRD</sub> | 45                       |      | 45    |      | ns   |
| InputData hold after SCLK rising  | t <sub>HSR</sub> | 0                        |      | 0     |      | ns   |

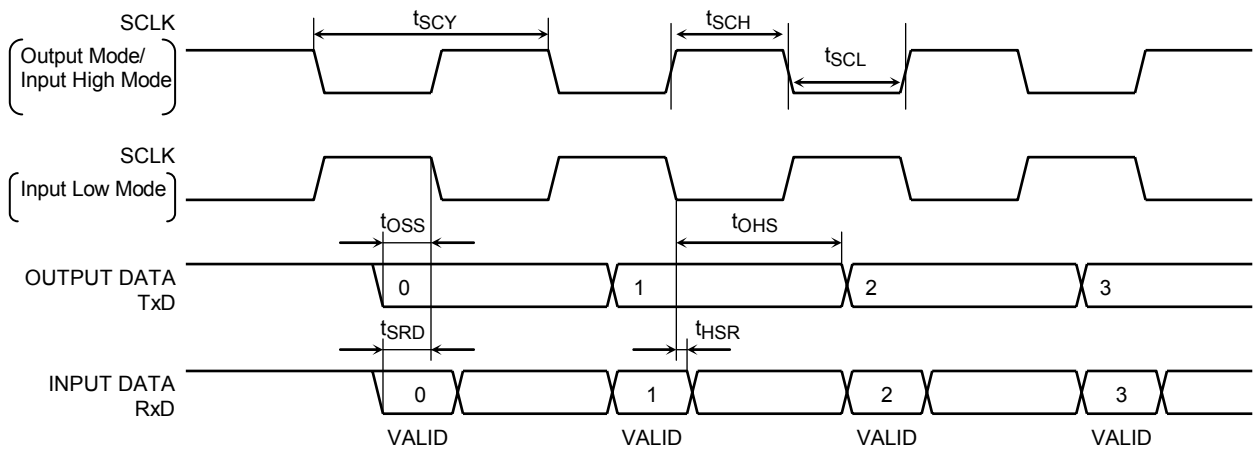


Fig 24-1 Serial channel timing(SIO)

### 24.7.2 Event Counter

Ta = -40 to 85°C

| 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   |

### 24.7.3 Capture

Ta = -40 to 85°C

| 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   |

### 24.7.4 General Interrupts

Ta = -40 to 85°C

| 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   |



**24.7.5 STOP Release Interrupts**

Ta = -40 to 85°C

| Parameter                      | Symbol             | Equation |      | 80MHz |      | Unit |
|--------------------------------|--------------------|----------|------|-------|------|------|
|                                |                    | Min.     | Max. | Min.  | Max. |      |
| Low pulse width for INT0 to F  | t <sub>INTBL</sub> | 100      |      | 100   |      | ns   |
| High pulse width for INT0 to F | t <sub>INTBH</sub> | 100      |      | 100   |      | ns   |

**24.7.6 Debug Communication**

(1) SWD Interface

| Parameter                   | Symbol           | Min. | Max. | Unit |
|-----------------------------|------------------|------|------|------|
| CLK cycle                   | T <sub>dck</sub> | 100  | -    | ns   |
| DATA hold after CLK rising  | T <sub>d1</sub>  | 4    | -    | ns   |
| DATA valid after CLK rising | T <sub>d2</sub>  | -    | 37   | ns   |
| DATA valid to CLK rising    | T <sub>ds</sub>  | 20   | -    | ns   |
| DATA hold after CLK falling | T <sub>dh</sub>  | 15   | -    | ns   |

(2) JTAG Interface

| Parameter                    | Symbol           | Min. | Max. | Unit |
|------------------------------|------------------|------|------|------|
| CLK cycle                    | T <sub>dck</sub> | 100  | -    | ns   |
| DATA hold after CLK falling  | T <sub>d3</sub>  | 4    | -    | ns   |
| DATA valid after CLK falling | T <sub>d4</sub>  | -    | 37   | ns   |
| DATA valid to CLK rising     | T <sub>ds</sub>  | 20   | -    | ns   |
| DATA hold after CLK rising   | T <sub>dh</sub>  | 15   | -    | ns   |

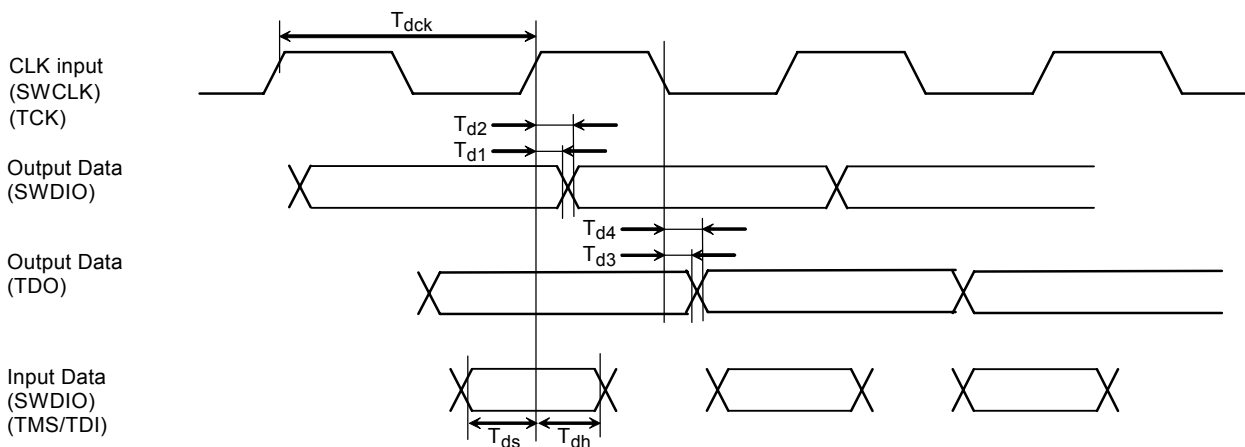


Fig 24-2 JTAG and SWD communication timing

### 24.7.7 TRACE Output

AC measurement 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               | $t_{clk}$    | 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   |

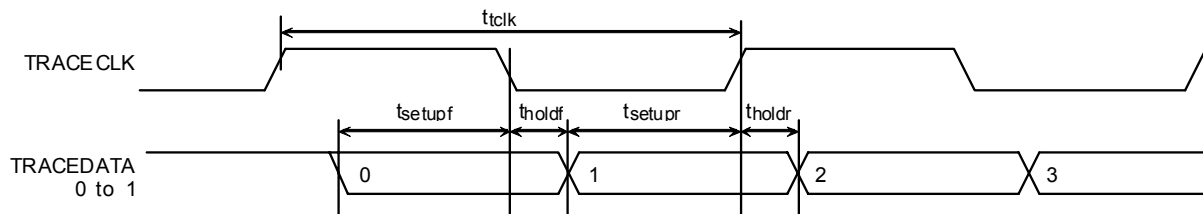


Fig 24-3 TRACE communication timing

### 24.8 Flash Characteristics

Ta = 0 to 70°C

| Parameter                      | Rating | Min. | Typ. | Max. | Unit  |
|--------------------------------|--------|------|------|------|-------|
| Flash memory erase/write times |        |      |      | 100  | times |

### 24.9 Oscillation Circuit

The TMPM370 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

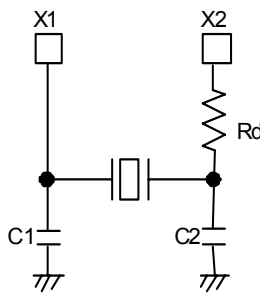


Fig 24-4 High-frequency oscillation connection

(2) Recommended ceramic oscillator

The TMPM370 recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd. Please refer to the following URL for details.  
<http://www.murata.co.jp>

### 24.10 Handling Precaution

#### Solderability

| Test parameter | Test condition  | Note  |
|----------------|---|---|
| Solderability  | Use of Sn-37Pb solder Bath<br>Solder bath temperature = 230 ° C, Dipping time = 5 seconds<br>The number of times = one, Use of R-type flux        | Pass:<br>solderability rate until forming · 95% |
|                | Use of Sn-3.0Ag-0.5Cu solder bath<br>Solder bath temperature = 245 ° C, Dipping time = 5 seconds<br>The number of times = one, Use of R-type flux |   |

### 24.11 Note the power on

Note for usage of Port L(PL0,PL1) when power on.  
 When power on, until VDDs reach operation voltage and passed 200µs, port L(PL0,PL1) must be OPEN or to supply "Low" level(less than 0.5V).  
 It is necessary to same measures that the power supply voltage dropped during operating, reset signal is generated by power on reset circuit, and power supply line rising again.  
 (Note) VDDs : DVDD5, DVDD5E, RVDD5, AVDD5A/B, AMPVDD5 must be supplied same voltage.

#### 24.11.1 Using Power On Reset only

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 |     | $2^{15}/f_{osc}$ |     | s    |
| $t_{DVDD}$ | Rising time of power line            |     |                  | 3   | ms   |

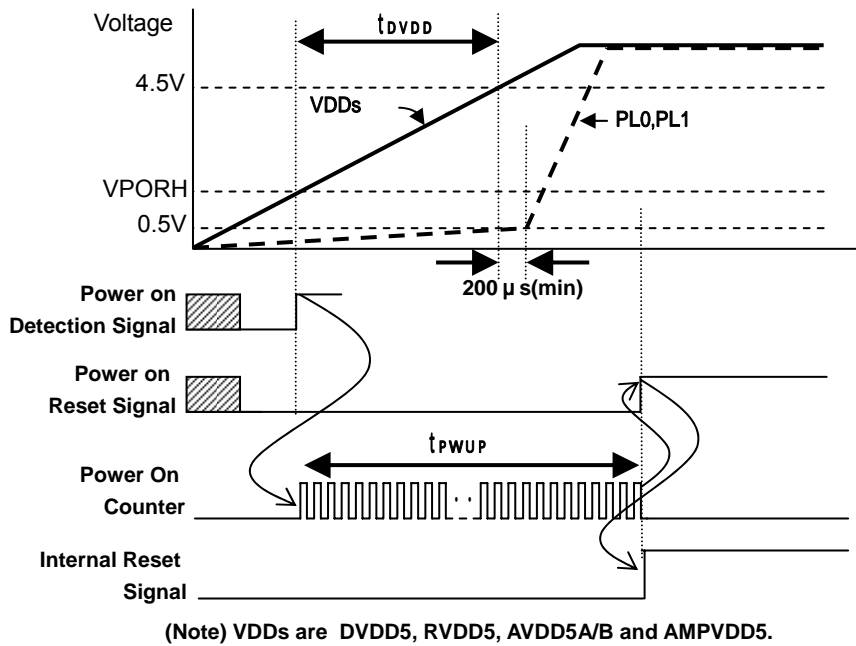


Fig 24.4 Power on Sequence(Using POR only)

24.11.2 Using External reset

(1) In case of the time of external reset shorter than POR

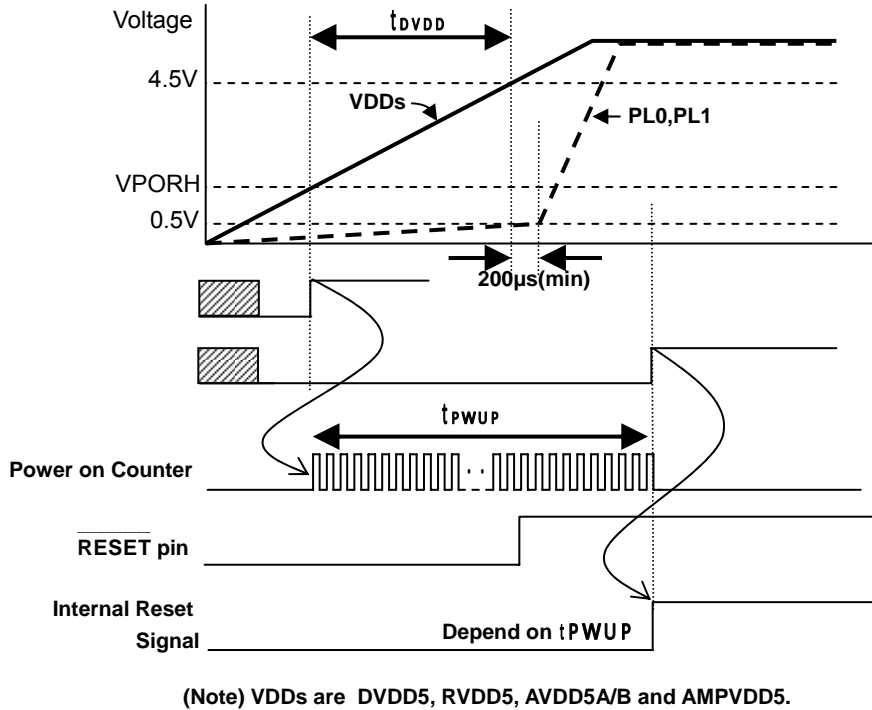


Fig 24.5 Power on Sequence(Using POR and External reset)(1)

(2) In case of the time of external reset longer than tPWUP

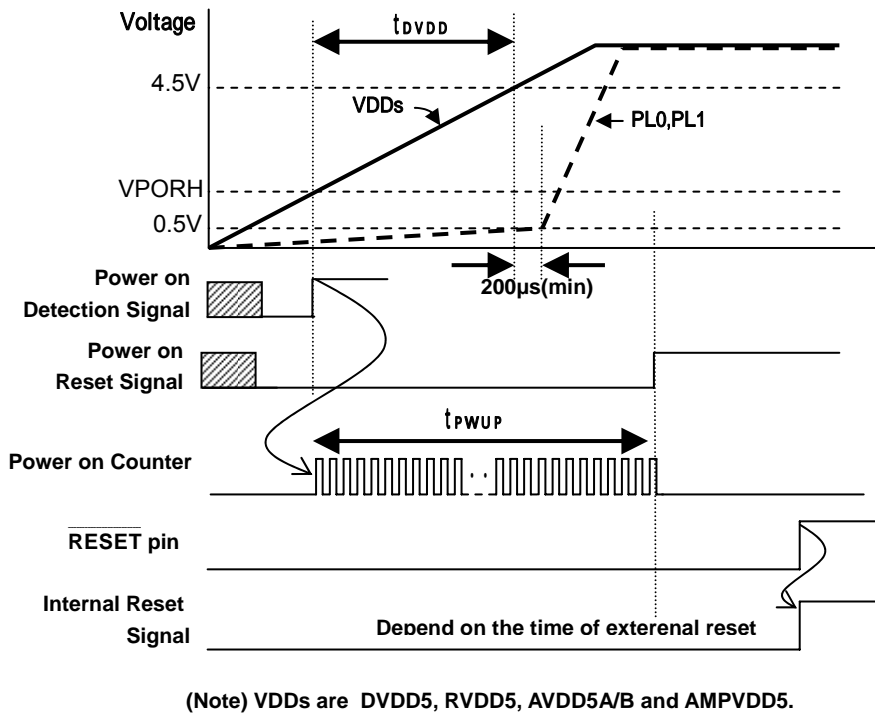
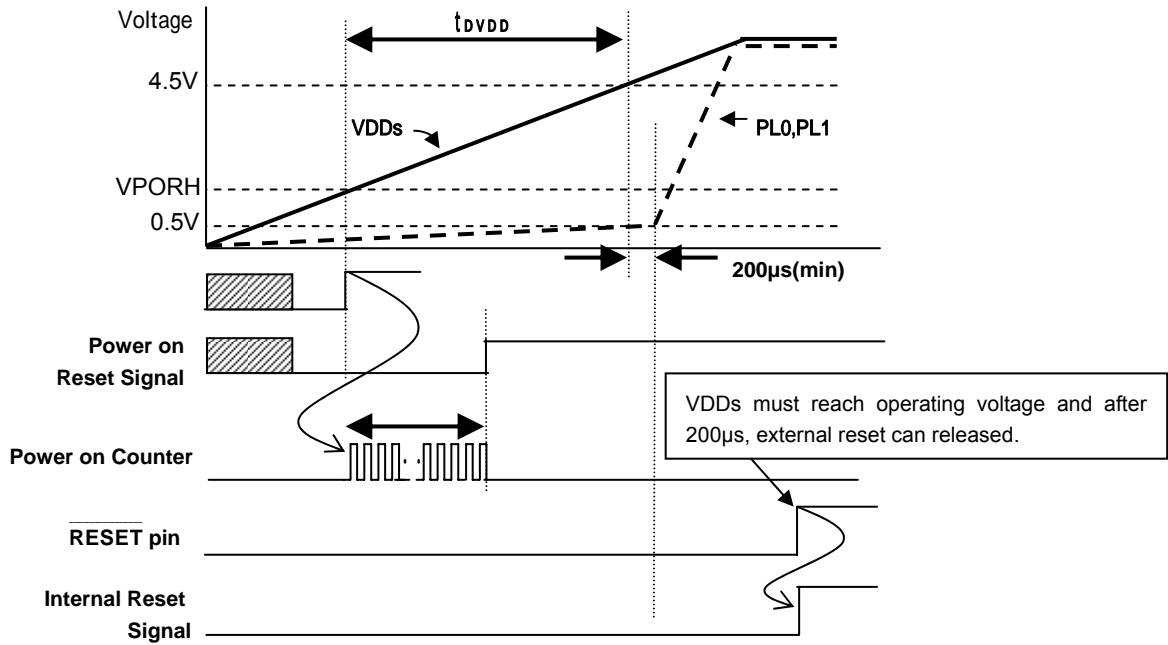


Fig 24.6 Power on Sequence(Using POR and External reset)(2)

(3) In case of the rising time of power line longer than  $t_{PWUP}$

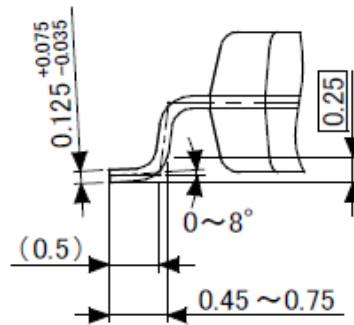
(Note) In this case, must be reset from  $\overline{RESET}$  pin.



(Note) VDDs are DVDD5, RVDD5, AVDD5A/B and AMPVDD5.

Fig 24.7 Power on Sequence ( $t_{DVDD} > t_{PWUP}$ )



Pin detail

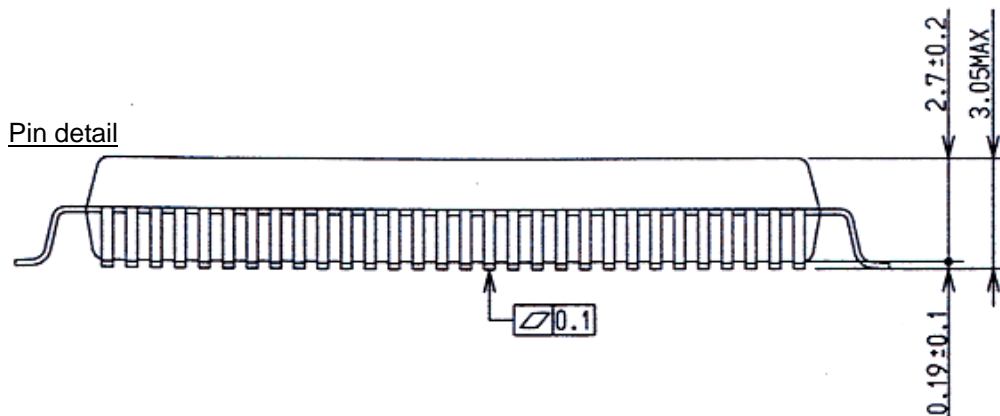
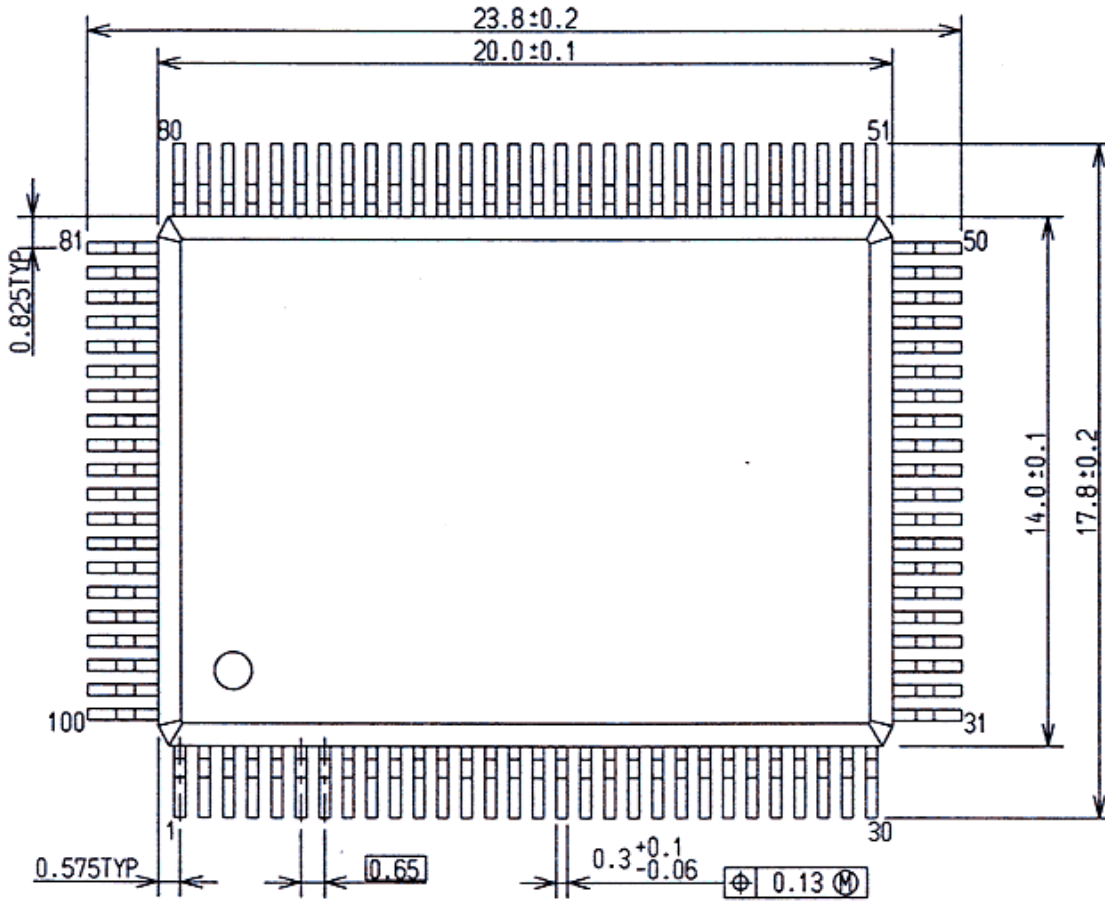
- (Note 1) For more dimensional information, please contact any one of our representatives  
(Note 2) The package is palladized



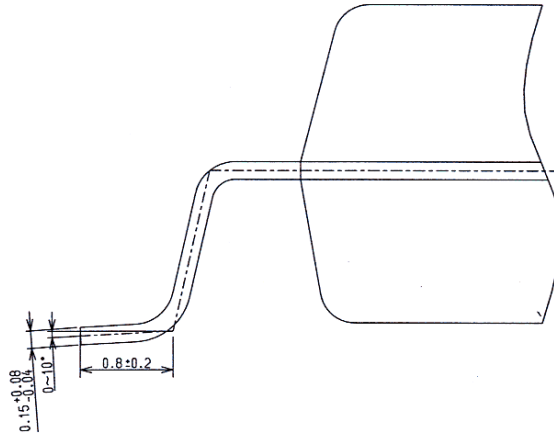
Type : QFP100-P-1420-0.65Q

Dimensions

Unit : mm



Pin detail



- (Note1) For more dimensional information, please contact any one of our representatives
- (Note2) The package is palladized

## RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact ("Unintended Use"). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.