TOSHIBA

32-bit RISC Microcontroller

TX03 Series

TMPM376FDDFG / TMPM376FDFG

TOSHIBA CORPORATION

Semiconductor & Storage Products Company

Revision History

Date	Rev	Description
2011/12/28	1.0	First Release
2013/4/12	2.0	Contents Revised

TMPM376FDDFG/FDFG

TMPM376FDDFG/FDFG is a 32-bit RISC microprocessor series with an ARM Cortex™-M3 microprocessor core.

Product Name	ROM (FLASH)	RAM	Package
TMPM376FDDFG	512 Kbyte	32 Kbyte	P-QFP100-1420-0.65Q
TMPM376FDFG	512 Kbyte	32 Kbyte	P-LQFP100-1414-0.50H

Features of the TMPM376FDDFG/FDFG are as follows:

1.1 Features

1. ARM Cortex-M3 microprocessor core

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

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

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

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

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

2. On Chip program memory and data memory
   - On-chip RAM : 32Kbyte
   - On-chip FlashROM : 512Kbyte

3. 16-bit timer (TMRB) : 8 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - Input capture function
   - External trigger PPG output

4. Watchdog timer (WDT) : 1 channel

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

5. Power_On reset function (POR)

6. Voltage detect function (VLTD)

7. Oscillation frequency detect function (OFD)

8. Vector engine (VE) : 1unit
   - Calculation circuit for motor control
   - Corresponding to 2 motors

9. Programmable motor driver (PMD) : 2channels
   - 3phase complementary PWM generator
   - Synchronous AD convert start trigger generator
   - Emergency protective function (EMG)

10. Encoder input circuit (ENC) : 2channels
    - 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. General-purpose serial interface(SIO/UART) : 4channels

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

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

    Either I2C bus mode or synchronous mode can be selected.

13. 12 bit AD converter (ADC) : 2units (Analog input : 22channel)
    - Start by the internal trigger : TMRB interrupt / PMD trigger
    - Constant conversion mode
    - AD monitoring 2ch
    - Conversion speed 2 μsec (@ADC conversion clock = 40 MHz)

14. Input/ output ports (PORT) : 82 pins

    I/O pin : 80 pins

    Input pin : 2 pins

15. Interrupt source
    - Internal 63 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.

16. Standby mode

Standby modes : 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. Internal high-speed oscillation circuit

20. Maximum operating frequency : 80 MHz

21. Operating voltage range

4.5 V to 5.5 V (with on-chip regulator)

22. Temperature range

- –40°C to 85°C (except during Flash writing/ erasing)
- 0°C to 70°C (during Flash writing/ erasing)

23. Package

- P-QFP100-1420-0.65Q (14 mm × 20 mm, 0.65 mm pitch)
- P-LQFP100-1414-0.50H (14 mm × 14 mm, 0.5 mm pitch)

1.2 Block Diagram

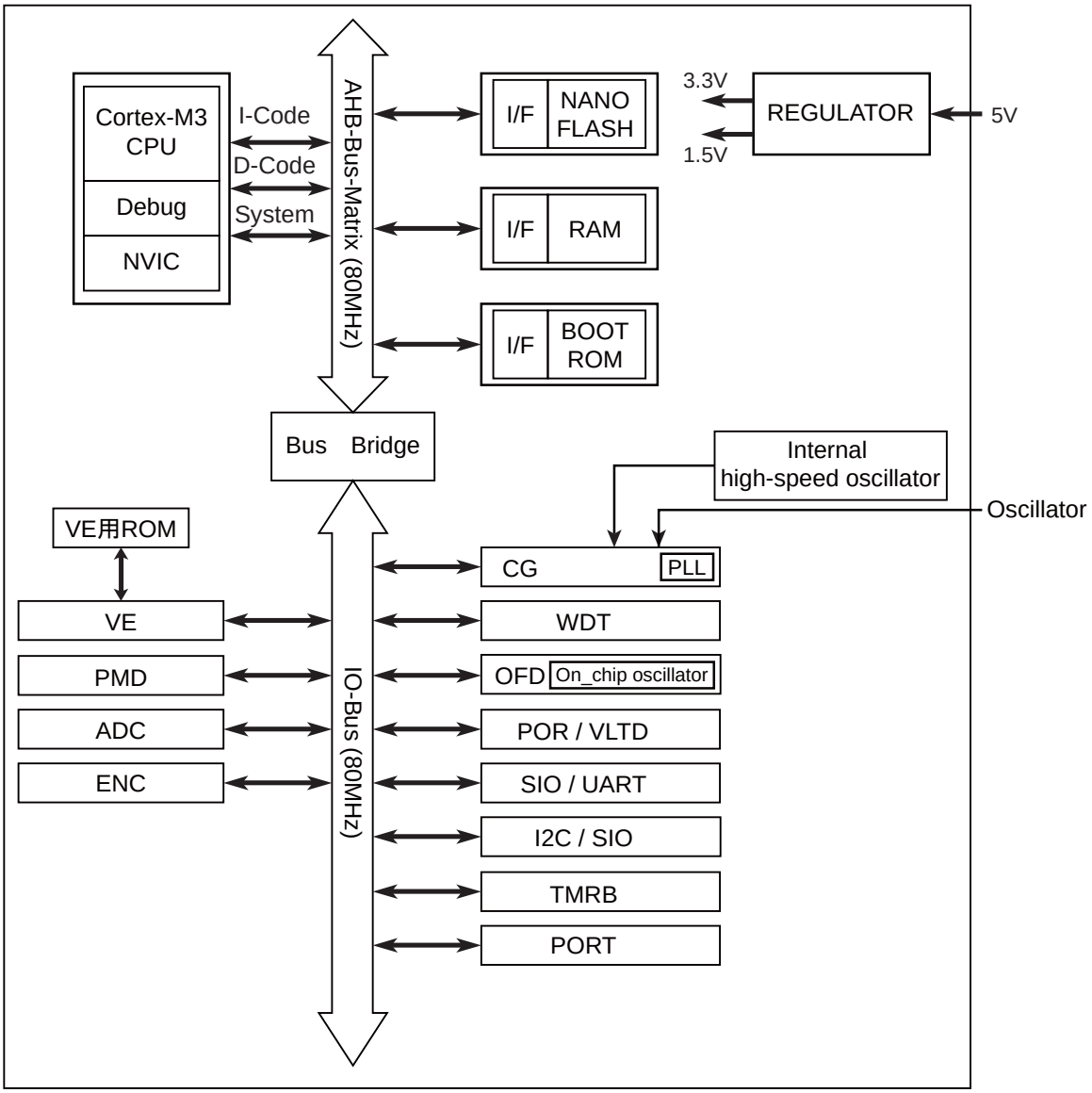


Figure 1-1 TMPM376FDDFG/FDFG block diagram

1.3 Pin Layout (Top view)

The pin layout of TMPM376FDDFG/FDFG is a figure below.

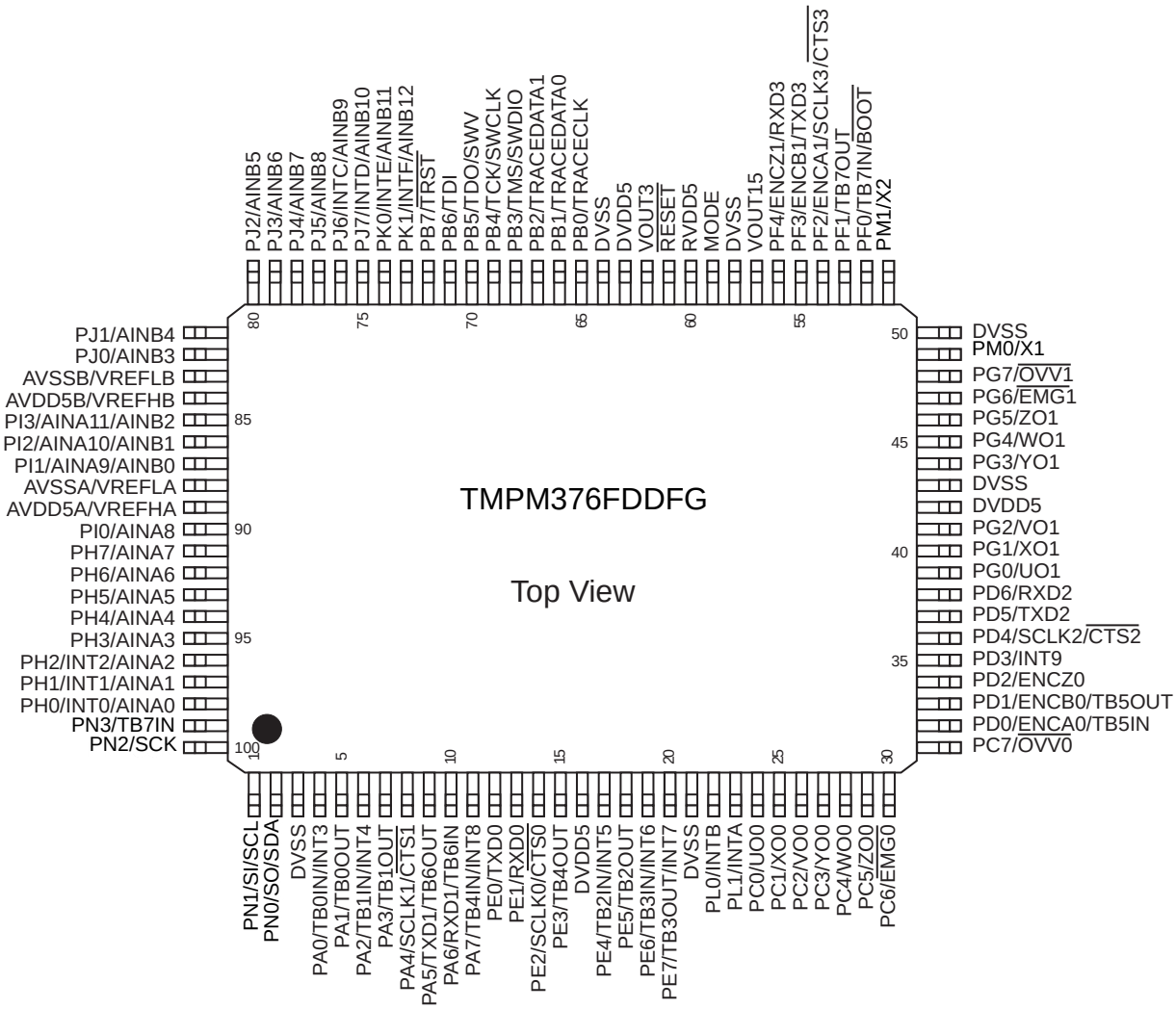


Figure 1-2 Pin Layout (QFP100)

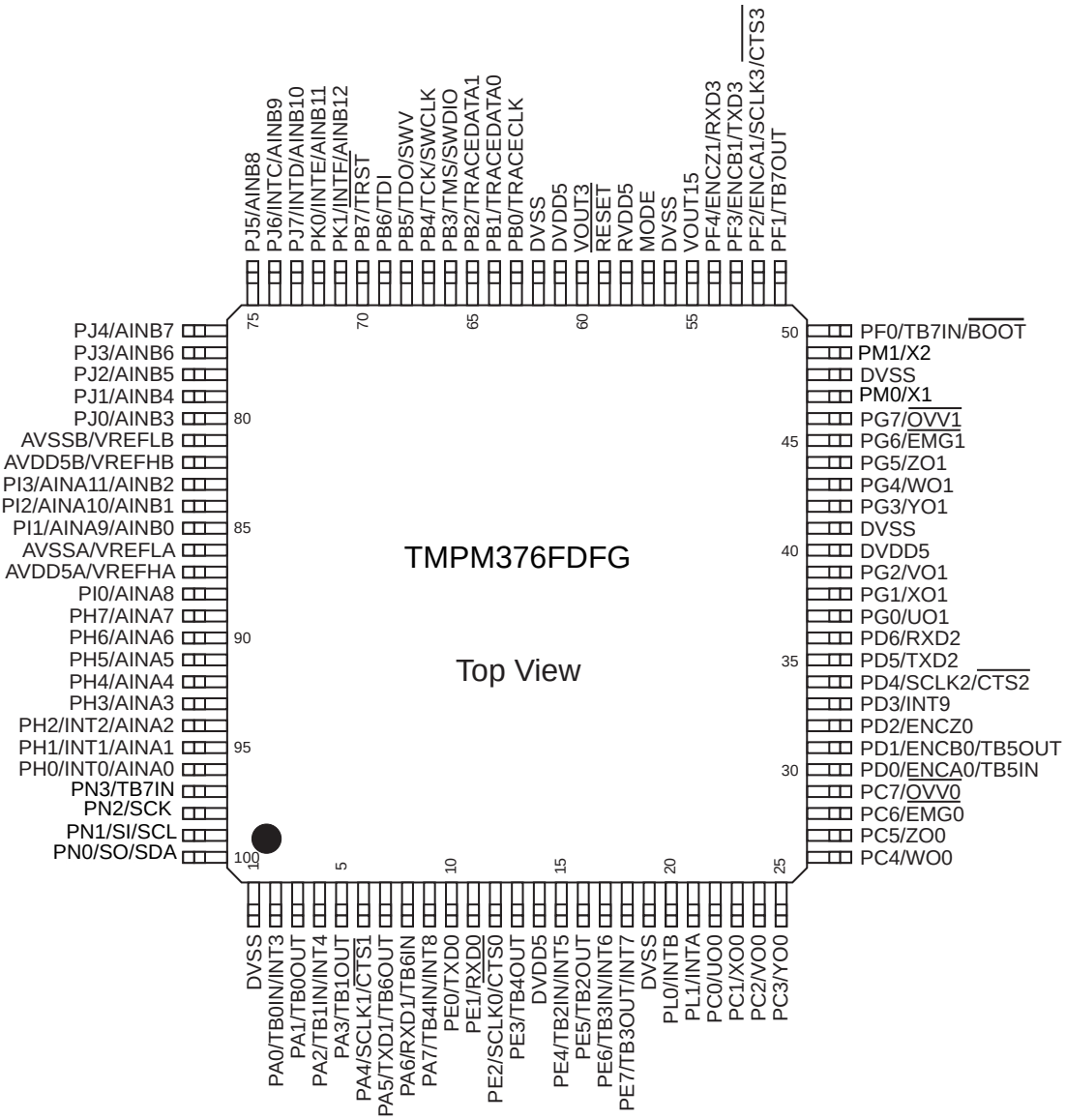


Figure 1-3 Pin Layout (LQFP100)

1.4 Pin names and Functions

Table 1-1 sorts the input and output pins of the TMPM376FDDFG/FDFG by pin or port. Each table includes alternate pin names and functions for multi-function pins.

1.4.1 Sorted by Port

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

PORT	Type	Pin No. (DFG/ FG)	Pin Name	Input / Output	Function
PORT A	Function	4 / 2	PA0 TB0IN INT3	I/O I I	I/O port Inputting the timer B capture trigger External interrupt pin
PORT A	Function	5 / 3	PA1 TB0OUT	I/O O	I/O port Timer B output
PORT A	Function	6 / 4	PA2 TB1IN INT4	I/O I I	I/O port Inputting the timer B capture trigger External interrupt pin
PORT A	Function	7 / 5	PA3 TB1OUT	I/O O	I/O port Timer B output
PORT A	Function	8 / 6	PA4 SCLK1 $\overline{\text{CTS1}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
PORT A	Function	9 / 7	PA5 TXD1 TB6OUT	I/O O O	I/O port Sending serial data Timer B output
PORT A	Function	10 / 8	PA6 RXD1 TB6IN	I/O I I	I/O port Receiving serial data Inputting the timer B capture trigger
PORT A	Function	11 / 9	PA7 TB4IN INT8	I/O I I	I/O port Inputting the timer B capture trigger External interrupt pin
PORT B	Function/ Debug	65 / 63	PB0 TRACECLK	I/O O	I/O port Debug pin
PORT B	Function/ Debug	66 / 64	PB1 TRACEDATA0	I/O O	I/O port Debug pin
PORT B	Function/ Debug	67 / 65	PB2 TRACEDATA1	I/O O	I/O port Debug pin
PORT B	Function/ Debug	68 / 66	PB3 TMS/SWDIO	I/O I/O	I/O port Debug pin
PORT B	Function/ Debug	69 / 67	PB4 TCK/SWCLK	I/O I	I/O port Debug pin
PORT B	Function/ Debug	70 / 68	PB5 TDO/SWV	I/O O	I/O port Debug pin
PORT B	Function/ Debug	71 / 69	PB6 TDI	I/O I	I/O port Debug pin
PORT B	Function/ Debug	72 / 70	PB7 $\overline{\text{TRST}}$	I/O I	I/O port Debug pin
PORT C	Function	24 / 22	PC0 UO0	I/O O	I/O port U-phase output pin

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

PORT	Type	Pin No. (DFG/ FG)	Pin Name	Input / Output	Function
PORT C	Function	25 / 23	PC1 XO0	I/O O	I/O port X-phase output pin
PORT C	Function	26 / 24	PC2 VO0	I/O O	I/O port V-phase output pin
PORT C	Function	27 / 25	PC3 YO0	I/O O	I/O port Y -phase output pin
PORT C	Function	28 / 26	PC4 WO0	I/O O	I/O port W-phase output pin
PORT C	Function	29 / 27	PC5 ZO0	I/O O	I/O port Z-phase output pin
PORT C	Function	30 / 28	PC6 $\overline{\text{EMG0}}$	I/O I	I/O port Emergency status detection input
PORT C	Function	31 / 29	PC7 $\overline{\text{OVV0}}$	I/O I	I/O port Overvoltage Detection Input
PORT D	Function	32 / 30	PD0 ENCA0 TB5IN	I/O I I	I/O port A-phase input pin Inputting the timer B capture trigger
PORT D	Function	33 / 31	PD1 ENCB0 TB5OUT	I/O I O	I/O port B-phase input pin Timer B output
PORT D	Function	34 / 32	PD2 ENCZ0	I/O I	I/O port Z-phase input pin
PORT D	Function	35 / 33	PD3 INT9	I/O I	I/O port External interrupt pin
PORT D	Function	36 / 34	PD4 SCLK2 $\overline{\text{CTS2}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
PORT D	Function	37 / 35	PD5 TXD2	I/O O	I/O port Sending serial data
PORT D	Function	38 / 36	PD6 RXD2	I/O I	I/O port Receiving serial data
PORT E	Function	12 / 10	PE0 TXD0	I/O O	I/O port Sending serial data
PORT E	Function	13 / 11	PE1 RXD0	I/O I	I/O port Receiving serial data
PORT E	Function	14 / 12	PE2 SCLK0 $\overline{\text{CTS0}}$	I/O I/O I	I/O port Serial clock input/ output Handshake input pin
PORT E	Function	15 / 13	PE3 TB4OUT	I/O O	I/O port Timer B output
PORT E	Function	17 / 15	PE4 TB2IN INT5	I/O I I	I/O port Inputting the timer B capture trigger External interrupt pin
PORT E	Function	18 / 16	PE5 TB2OUT	I/O O	I/O port Timer B output
PORT E	Function	19 / 17	PE6 TB3IN INT6	I/O I I	I/O port Inputting the timer B capture trigger External interrupt pin

Table 1-1 Pin Names and Functions Sorted by Port (3/5)

PORT	Type	Pin No. (DFG/ FG)	Pin Name	Input / Output	Function
PORTE	Function	20 / 18	PE7 TB3OUT INT7	I/O O I	I/O port Timer B output External interrupt pin
PORT F	Function/ Control	52 / 50	PF0 TB7IN $\overline{\text{BOOT}}$	I/O I I	I/O port Inputting the timer B capture trigger (Note 3) BOOT mode pin. (note) This pin goes into single boot mode by sampling "Low" at the rise of a RESET signal.
PORT F	Function	53 / 51	PF1 TB7OUT	I/O O	I/O port Timer B output
PORT F	Function	54 / 52	PF2 ENCA1 SCLK3 $\overline{\text{CTS3}}$	I/O I I/O I	I/O port Encoder Input Serial clock input/ output Handshake input pin
PORT F	Function	55 / 53	PF3 ENCB1 TXD3	I/O I O	I/O port Encoder Input Sending serial data
PORT F	Function	56 / 54	PF4 ENCZ1 RXD3	I/O I I	I/O port Encoder Input Receiving serial data
PORT G	Function	39 / 37	PG0 UO1	I/O O	I/O port U-phase output pin
PORT G	Function	40 / 38	PG1 XO1	I/O O	I/O port X-phase output pin
PORT G	Function	41 / 39	PG2 VO1	I/O O	I/O port V-phase output pin
PORT G	Function	44 / 42	PG3 YO1	I/O O	I/O port Y-phase output pin
PORT G	Function	45 / 43	PG4 WO1	I/O O	I/O port W-phase output pin
PORT G	Function	46 / 44	PG5 ZO1	I/O O	I/O port Z-phase output pin
PORT G	Function	47 / 45	PG6 $\overline{\text{EMG1}}$	I/O I	I/O port Emergency status detection input
PORT G	Function	48 / 46	PG7 $\overline{\text{OVV1}}$	I/O I	I/O port Overvoltage Detection Input
PORT H	Function	98 / 96	PH0 INT0 AINA0	I/O I I	I/O port External interrupt pin Analog input
PORT H	Function	97 / 95	PH1 INT1 AINA1	I/O I I	I/O port External interrupt pin Analog input
PORT H	Function	96 / 94	PH2 INT2 AINA2	I/O I I	I/O port External interrupt pin Analog input
PORT H	Function	95 / 93	PH3 AINA3	I/O I	I/O port Analog input
PORT H	Function	94 / 92	PH4 AINA4	I/O I	I/O port Analog input

Table 1-1 Pin Names and Functions Sorted by Port (4/5)

PORT	Type	Pin No. (DFG/ FG)	Pin Name	Input / Output	Function
PORT H	Function	93 / 91	PH5 AINA5	I/O I	I/O port Analog input
PORT H	Function	92 / 90	PH6 AINA6	I/O I	I/O port Analog input
PORT H	Function	91 / 89	PH7 AINA7	I/O I	I/O port Analog input
PORT I	Function	90 / 88	PI0 AINA8	I/O I	I/O port Analog input
PORT I	Function	87 / 85	PI1 AINA9/AINB0	I/O I	I/O port Analog input
PORT I	Function	86 / 84	PI2 AINA10/AINB1	I/O I	I/O port Analog input
PORT I	Function	85 / 83	PI3 AINA11/AINB2	I/O I	I/O port Analog input
PORT J	Function	82 / 80	PJ0 AINB3	I/O I	I/O port Analog input
PORT J	Function	81 / 79	PJ1 AINB4	I/O I	I/O port Analog input
PORT J	Function	80 / 78	PJ2 AINB5	I/O I	I/O port Analog input
PORT J	Function	79 / 77	PJ3 AINB6	I/O I	I/O port Analog input
PORT J	Function	78 / 76	PJ4 AINB7	I/O I	I/O port Analog input
PORT J	Function	77 / 75	PJ5 AINB8	I/O I	I/O port Analog input
PORT J	Function	76 / 74	PJ6 INTC AINB9	I/O I I	I/O port External interrupt pin Analog input
PORT J	Function	75 / 73	PJ7 INTD AINB10	I/O I I	I/O port External interrupt pin Analog input
PORT K	Function	74 / 72	PK0 INTE AINB11	I/O I I	I/O port External interrupt pin Analog input
PORT K	Function	73 / 71	PK1 INTF AINB12	I/O I I	I/O port External interrupt pin Analog input
PORT L	Function	22 / 20	PL0 INTB	I I	Input port External interrupt pin
PORT L	Function	23 / 21	PL1 INTA	I I	Input port External interrupt pin
PORT M	Function / Clock	49 / 47	PM0 X1	I/O I	I/O port Connected to a high-speed oscillator
PORT M	Function / Clock	51 / 49	PM1 X2	I/O O	I/O port Connected to a high-speed oscillator

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

PORT	Type	Pin No. (DFG/ FG)	Pin Name	Input / Output	Function
PORT N	Function	2 / 100	PN0 SO / SDA	I/O I/O	I/O port If the serial bus interface operates -in the SIO mode: data pin / -in the I2C mode: data pin
PORT N	Function	1 / 99	PN1 SI / SCL	I/O I/O	I/O port If the serial bus interface operates -in the SIO mode: data pin / -in the I2C mode: clock pin
PORT N	Function	100 / 98	PN2 SCK	I/O I/O	I/O port Inputting and outputting a clock if the serial bus interface operates in the SIO mode.
PORT N	Function	99 / 97	PN3 TB7IN	I/O I	I/O port Inputting the timer B capture trigger (NOTE3)
-	Control	59 / 57	MODE	I	Mode pin (note) MODE pin must be connected to GND.
-	Function	61 / 59	$\overline{\text{RESET}}$	I	Reset input pin (note) With a pull-up and a noise filter (about 30ns (typical value))
-	PS	3 / 1	DVSS	–	GND pin
-	PS	21 / 19	DVSS	–	GND pin
-	PS	43 / 41	DVSS	–	GND pin
-	PS	50 / 48	DVSS	–	GND pin
-	PS	58 / 56	DVSS	–	GND pin
-	PS	64 / 62	DVSS	–	GND pin
-	PS	16 / 14	DVDD5	–	Power supply pin
-	PS	42 / 40	DVDD5	–	Power supply pin
-	PS	63 / 61	DVDD5	–	Power supply pin
-	PS	60 / 58	RVDD5	–	Power supply pin
-	PS	57 / 55	VOUT15	–	Power supply pin
-	PS	62 / 60	VOUT3	–	Power supply pin
-	PS	83 / 81	AVSSB VREFLB	–	AD converter: GND pin (Note 1) Supplying the AD converter with a reference power supply.
-	PS	84 / 82	AVDD5B VREFHB	–	Supplying the AD converter with a power supply. (Note2) Supplying the AD converter with a reference power supply.
-	PS	88 / 86	AVSSA VREFLA	–	AD converter: GND pin (Note 1) Supplying the AD converter with a reference power supply.
-	PS	89 / 87	AVDD5A VREFHA	–	Supplying the AD converter with a power supply. (Note2) Supplying the AD converter with a reference power supply.

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

Note 2: Must be connected to power supply even if AD converter is not used.

Note 3: TB7IN cannot be used simultaneously.

1.5 Pin Numbers and Power Supply Pins

Table 1-2 Pin Numbers and Power Supplies

Power supply	Voltage range	Pin No. (DFG / FG)	Pin name
DVDD5	4.5 to 5.5V	16 / 14 , 42 / 40 , 63 / 61	PA,PB,PC,PD,PE,PF,PG,PL,PM PN,$\overline{\text{RESET}}$,MODE
AVDD5A		89 / 87	PH, PI
AVDD5B		84 / 82	PJ, PK
RVDD5		60 / 58	–
VOUT15	1.35 to 1.65V	57 / 55	VOUT15 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit.
VOUT3	2.7 to 3.6V	62 / 60	VOUT3 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit.

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

2. Processor Core

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

2.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM376FDDFG/FDFG.

Refer to the detailed information about the CPU core and architecture, refer to the ARM manual "Cortex-M series processors" in the following URL:

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

Product Name	Core Revision
TMPM376FDDFG/ FDFG	r2p0

2.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r2p0 are ETM™ and MPU. The following tables shows the configurable options in the TMPM376FDDFG/FDFG.

Configurable Options	Implementation
FPB	Two literal comparators Six instruction comparators
DWT	Four comparators
ITM	Implementable
MPU	Not implementable
ETM	Implementable
AHB-AP	Implementable
AHB Trace Macrocell Interface	Implementable
TPIU	Implementable
WIC	Not implementable

2.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

2.3.1 Number of Interrupt Inputs

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

TMPM376FDDFG/FDFG has 79 interrupt inputs. The number of interrupt inputs is reflected in <INTLINESNUM[4:0]> bit of NVIC register. In this product, if read <INTLINESNUM[4:0]> bit, 0x00 is read out.

2.3.2 Number of Priority Level Interrupt Bits

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

TMPM376FDDFG/FDFG has 3 priority level interrupt bits. The number of priority level interrupt bits is used for assigning a priority level in the interrupt priority registers and system handler priority registers.

2.3.3 SysTick

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

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

2.3.4 SYSRESETREQ

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

TMPM376FDDFG/FDFG provides the same operation when SYSRESETREQ signal are output.

2.3.5 LOCKUP

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

TMPM376FDDFG/FDFG does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interruput (NMI) or reset.

2.3.6 Auxiliary Fault Status register

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

However, TMPM376FDDFG/FDFG is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

2.4 Events

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

TMPM376FDDFG/FDFG does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

2.5 Power Management

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

These signals are output in the following circumstances:

-Wait-For-Interrupt (WFI) instruction execution

-Wait-For-Event (WFE) instruction execution

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

TMPM376FDDFG/FDFG does not use SLEEPDEEP signal so that <SLEEPDEEP> bit must not be set. And also event signal is not used so that please do not use WFE instruction.

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

2.6 Exclusive access

In Cortex-M3 core, the DCode bus system supports exclusive access. However TMPM376FDDFG/FDFG does not use this function.

3. Memory Map

3.1 Memory Map

The memory maps for MPM376FDDFG/FDFG are based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of MPM376FDDFG/FDFG are mapped to the Code, SRAM and peripheral regions of the Cortex-M3 respectively. The special function register (SFR) means the control registers of all input/output ports and peripheral functions. The SRAM and SFR areas are all included in the bit-band region.

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

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

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

3.1.1 MPM376FDDFG/FDFG Memory Map

Figure 3-1 shows the memory map of the MPM376FDDFG/FDFG.

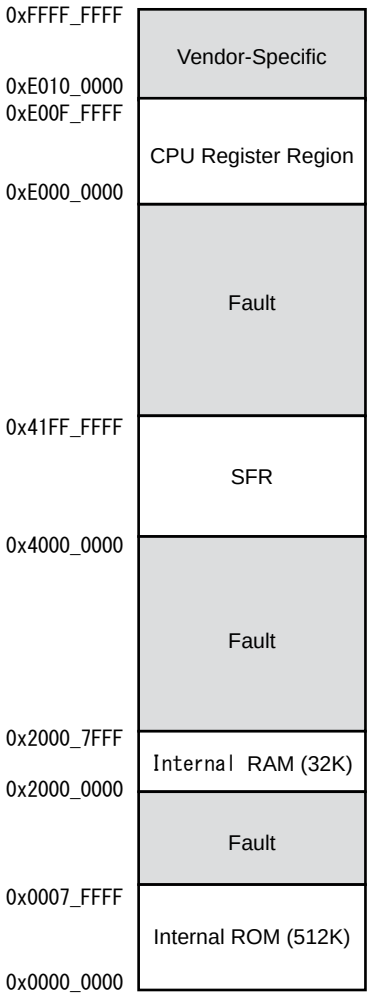


Figure 3-1 Memory Map

3.2 Details of SFR area

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

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

Table 3-1 Details of SFR

Start Address	End Address	Peripheral
0x4000_0000	0x4000_037F	PORT
0x4000_0380	0x4000_FFFF	Reserved
0x4001_0000	0x4001_01FF	TMRB
0x4001_0200	0x4001_03FF	Reserved
0x4001_0400	0x4001_053F	ENC
0x4001_0540	0x4001_FFFF	Reserved
0x4002_0000	0x4002_007F	I2C/SIO
0x4002_0080	0x4002_017F	SIO/UART
0x4002_0180	0x4002_FFFF	Reserved
0x4003_0000	0x4003_02FF	ADC
0x4003_0300	0x4003_FFFF	Reserved
0x4004_0000	0x4004_003F	WDT
0x4004_0040	0x4004_01FF	Reserved
0x4004_0200	0x4004_023F	CG
0x4004_0240	0x4004_07FF	Reserved
0x4004_0800	0x4004_083F	OFD
0x4004_0840	0x4004_08FF	Reserved
0x4004_0900	0x4004_093F	VLTD
0x4004_0940	0x4004_FFFF	Reserved
0x4005_0000	0x4005_023F	VE
0x4005_0240	0x4005_03FF	Reserved
0x4005_0400	0x4005_04FF	PMD
0x4005_0500	0x4007_FFFF	Reserved
0x4008_0000	0x41FF_EFFF	Hard fault
0x41FF_F000	0x41FF_F03F	FLASH
0x41FF_F040	0x41FF_FFFF	Reserved

4. Reset operation

4.1 Initial state

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

4.2 Reset operation

TMPM376FDDFG/FDFG has Power-on reset circuit, power-on reset signal is generated when power supply is turned on. When reset from external RESET pin, input reset signal to RESET pin at "Low" level for minimum duration of 1.2μsec while power supply voltage is in the operating range.

4.3 After Reset

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

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

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

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

5. Clock / Mode Control

5.1 Features

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

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

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

The clock/mode control block has the following functions:

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

In addition to NORMAL mode, the TMPM376FDDFG/FDFG can operate in six types of low power mode to reduce power consumption according to its usage conditions.

5.2 Registers

5.2.1 Register List

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

Base Address = 0x4004_0200

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

5.2.2 CGSYSCR (System control register)

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

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

5.2.3 CGOSCCR (Oscillation control register)

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

Bit	Bit Symbol	Type	Function
31-20	WUODR[11:0]	R/W	Specifies count time of the warm-up timer.
19	WUPSEL2	R/W	Clock source for Warm-up timer(WUP) 0: Internal (OSC2) 1: External (OSC1) Select source clock for warm-up timer between external oscillator (OSC1) and internal oscillator (OSC2).
18	HOSCON	R/W	Port M or external oscillator (X1/X2) (Note1) 0: PORT M 1: External oscillator (X1/X2) Specifies Port M or X1/X2. When the external oscillator (OSC1) is used, Port M registers (PMCR/PMPUP/PMPDN/PMIE) should be disabled. After reset, the port M registers are disabled.
17	OSCSEL	R/W	Selection of high-speed oscillator 0: Internal (OSC2) 1: External (OSC1) Select high-speed oscillator between external oscillator (OSC1) and internal oscillator (OSC2). Confirm <OSCSEL> become "1" then halt the OSC2 immediately after switching over to OSC1. Do not select OSC2 again after switching to OSC1.
16	XEN2	R/W	High-speed oscillator2 (Internal) 0: Stop 1:Oscillation Specifies operation of the high-speed oscillator 2 (OSC2).
15-12	–	R/W	Write as "0".
11-10	–	R	Read as "0".
9	–	R/W	Write as "0".
8	XEN1	R/W	High-speed oscillator1 (External) 0: Stop 1:Oscillation Specifies operation of the high-speed oscillator 1 (OSC1).
7-4	–	R/W	Read as "0".
3	WUPSEL1	R/W	Clock source for Warm-up timer Write as "0".
2	PLLON	R/W	PLL operation 0: Stop 1: Oscillation Specifies operation of the PLL. It stops after reset.Setting the bit is required.
1	WUEF	R	Status of warm-up timer (WUP) (Note2) 0: Warm-up completed. 1: Warm-up operation Enable to monitor the status of the warm-up timer.
0	WUEON	W	Operation of warm-up timer (Note2) 0: don't care 1: Starting warm-up Enables to start the warm-up timer.

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

Note 2: Do not write "1" to <WUEON>, at the setting of returning from stop mode with automatic warming-up. When warming-up is started by software (<WUOEN> = "1"), please monitor <WUEF> and confirm warming-up is completed. After <WUEN> turn to "0" operation mode can be changed to stop mode.

5.2.4 CGSTBYCR (Standby control register)

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

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

5.2.5 CGPLLSEL (PLL Selection Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-12	–	R/W	Write as "1010".
11	–	R	Read as "0".
10-1	–	R/W	Write as "00_1001_1111".
0	PLLSEL	R/W	Use of PLL 0: fosc use 1: PLL Use Specifies use or disuse of the clock multiplied by the PLL. "fosc" is automatically set after reset. Resetting is required when using the PLL.

5.3 Clock control

5.3.1 Clock Type

Each clock is defined as follows :

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

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

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

5.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

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

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

$f_C = f_{OSC2}$

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

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

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

5.3.3 Clock system Diagram

Figure 5-1 shows the clock system diagram.

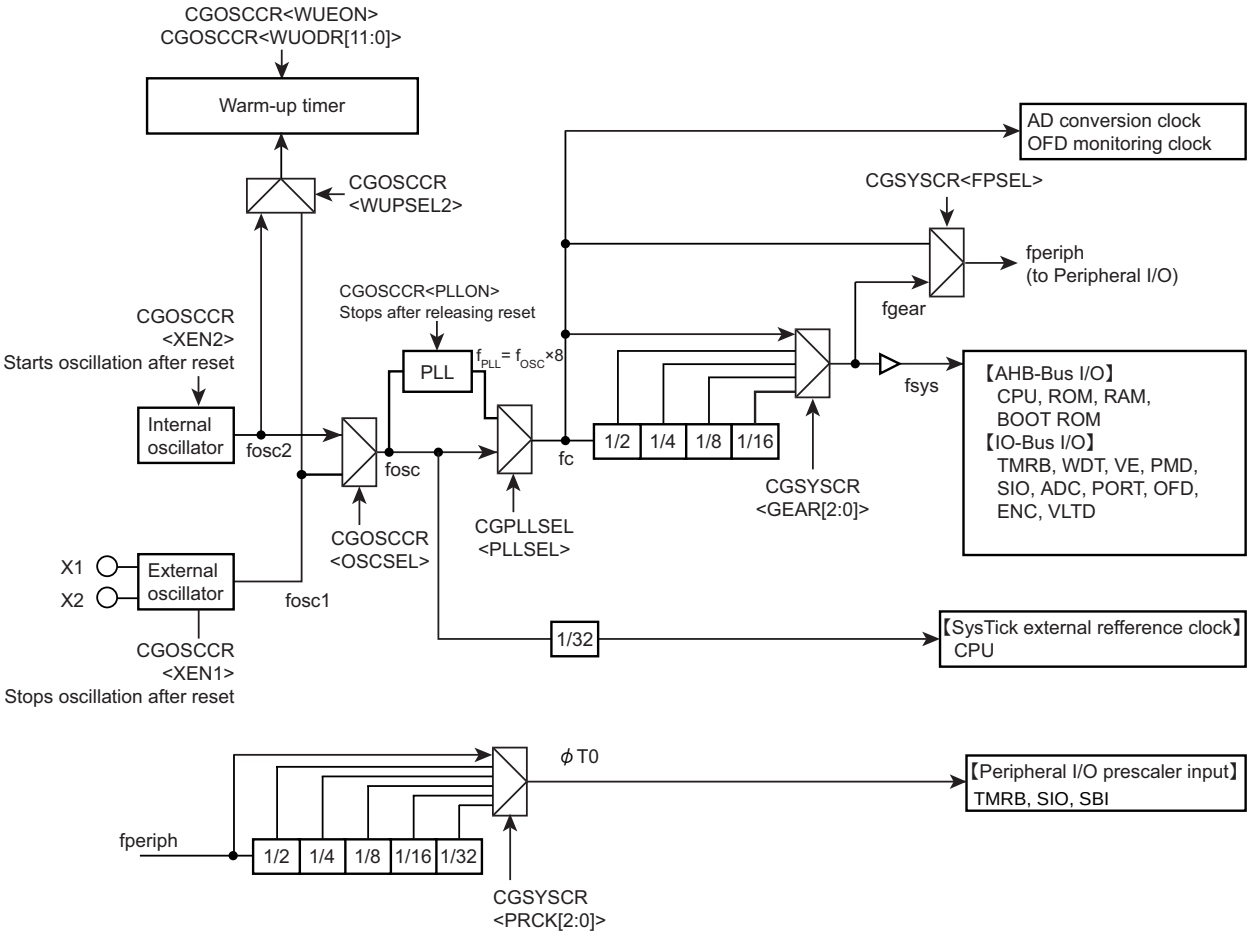


Figure 5-1 Clock Block Diagram

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

5.3.4 Clock Multiplication Circuit (PLL)

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

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

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

Note: It takes approximately 200μs for the PLL to be stabilized.

5.3.4.1 The sequence of PLL setting

The following shows PLL setting sequence after reset.

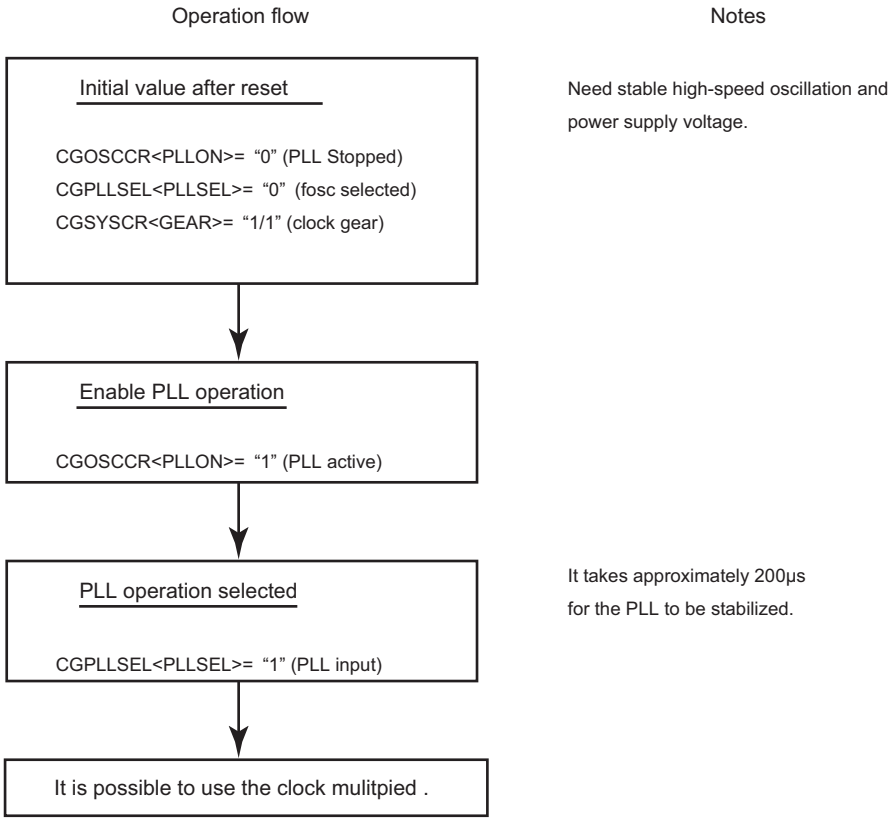


Figure 5-2 PLL setting sequence after reset

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

5.3.5 Warm-up function

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

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

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

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

How to configure the warm-up function.

1. Specify the count up clock

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

2. Specify the warm-up counter value

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

   The following shows the warm-up setting and example.

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

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

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

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

3. Confirm the start and completion of warm-up

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

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

The following shows the warm-up setting.

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

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

5.3.6 System Clock

The TMPM376FDDFG/FDFG offers high-speed clock as system clock. System clock is selectable from internal oscillator or external oscillator. After reset, internal oscillator is enabled and external oscillator is disabled. The high-speed clock is dividable.

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

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

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

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

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

Note 3: ."-" : Reserved

Note 4: Do not use 1/16 when "PLL =OFF" is used.

Note 5: Do not use 1/16 when SysTick is used.

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

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

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

5.3.7 Prescaler Clock Control

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

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

5.4 Modes and Mode Transitions

5.4.1 Mode Transitions

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

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

Figure 5-3 shows mode transition diagram.

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

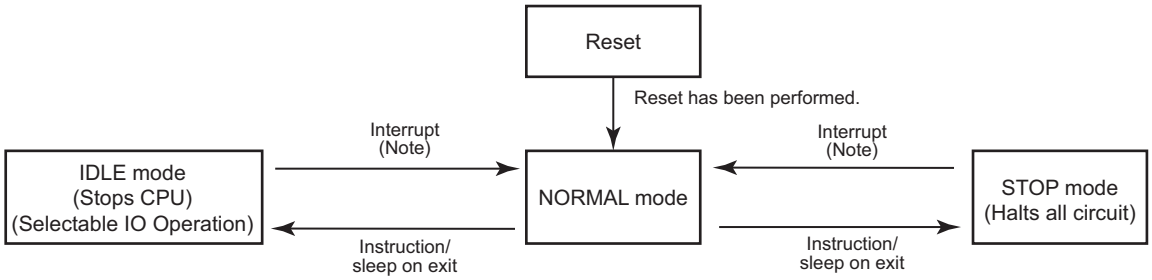


Figure 5-3 Mode Transition Diagram

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

5.5 Operation Mode

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

5.5.1 NORMAL mode

This mode is to operate the CPU core and the peripheral hardware by using the high-speed clock.

It is shifted to the NORMAL mode after reset.

5.6 Low Power Consumption Modes

The TMPM376FDDFG/FDFG has two low power consumption modes: IDLE and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction.In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

Note 1: The TMPM376FDDFG/FDFG does not offer any event for releasing the low power consumption mode. Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.

Note 2: The TMPM376FDDFG/FDFG does not support the low power consumption mode configured with the SLEEP-DEEP bit in the Cortex-M3 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

The features of each mode are described as follows.

5.6.1 IDLE Mode

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

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

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

Note: WDT should be stopped before entering IDLE mode.

5.6.2 STOP mode

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

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

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

Table 5-2 Pin States in the STOP mode

	Pin name	I/O	<DRVE> = 0	<DRVE> = 1
Not port	$\overline{\text{RESET}}$, MODE	Input only	o	
Port	X1	Input only	×	
	X2	Output only	"High" level output	
	TMS TCK TDI $\overline{\text{TRST}}$	Input	o	
	TDO	Output	Enabled when data is valid. Disabled when data is invalid.	
	SWCLK	Input	o	
	SWDIO	Input	o	
		Output	Enabled when data is valid. Disabled when data is invalid.	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	Output	o	
	UO0,1 VO0,1 WO0,1 XO0,1 YO0,1 ZO0,1	Output	Enabled when data is valid. Disabled when data is invalid.	
	INT0, INT1, INT2 INT3, INT4, INT5 INT6, INT7, INT8 INT9, INTA, INTB INTC, INTD, INTE INTF	Input	o	
	Other function pins other than the above or the ports that are used as general purpose ports.	Input	×	o
		Output	×	o

o : Input or output enabled.
× : Input or output disabled.

5.6.3 Low power Consumption Mode Setting

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

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

Table 5-3 Low power consumption mode setting

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

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

5.6.4 Operational Status in Each Mode

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

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

Table 5-4 Operational Status in Each Mode

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

o : Operating

× : Stopped

Note 1: It depends on CGSTBYCR<DRVE>.
Note 2: The blocks are not stopped even though the clock is halted.

5.6.5 Releasing the Low Power Consumption Mode

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

Details are shown in Table 5-5.

Table 5-5 Release Source in Each Mode

Low power consumption mode			IDLE (programable)	STOP
Release source	Interrupt	INT0 to F (Note1)	o	o
		INTRX0 to 3, INTTX0 to 3	o	×
		INTVCNA, INTVCNB	o	×
		INTEMG0 to 1	o	×
		INTOVV0 to 1	o	×
		INTADAPDA, INTADBPDA, INTADAPDB, INTADBPDB	o	×
		INTTB00, 10, 20, 30, 40, 50, 60, 70 INTTB01, 11, 21, 31, 41, 51, 61, 71	o	×
		INTPMD0, 1	o	×
		INTCAP00, 10, 20, 30, 40, 50, 60, 70 INTCAP01, 11, 21, 31, 41, 51, 61, 71	o	×
		INTADACPA, INTADBCPA, INTADACPB, INTADBCPB	o	×
		INTADASFT, INTADBSFT	o	×
		INTADATMR, INTADBTMR	o	×
		INTENC0, INTENC1	o	×
		INTSBI	o	×
	SysTick		o	×
	NMI (INTWDT)		o	×
	RESET ($\overline{\text{RESET}}$ pin)		o	o

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

Note 1: To release the low power consumption mode by using the level mode interrupt, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt handling from starting properly.

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

Note 3: Refer to "5.6.6 Warm-up" about warm-up time.

- Release by interrupt request

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

- Release by Non-Maskable Interrupt (NMI)

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

- Release by reset

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

- Release by SysTick interrupt

  SysTick interrupt can only be used in IDLE mode.

Refer to "Interrupts" for detail.

5.6.6 Warm-up

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

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

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

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

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

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

Table 5-6 Warm-up setting in mode transition

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

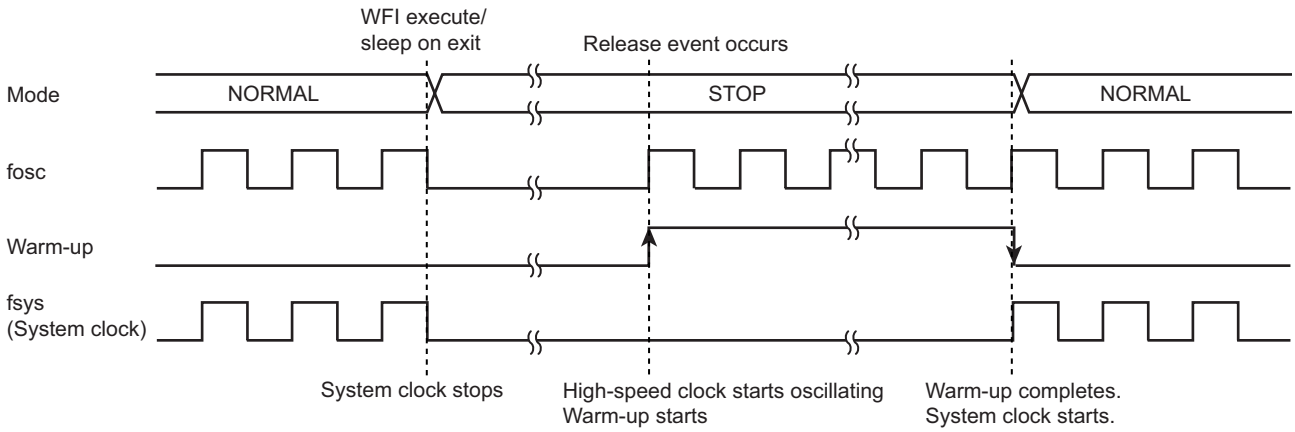
5.6.7 Clock Operation in Mode Transition

The clock operation in mode transition are described Chapter 5.6.7.1.

5.6.7.1 Transition of operation modes : NORMAL → STOP → NORMAL

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

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



6. Exceptions

This chapter describes features, types and handling of exceptions.

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

6.1 Overview

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

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

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

6.1.1 Exception types

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

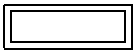
For detailed descriptions on each exception, refer to "Cortex-M3 Technical Reference Manual".

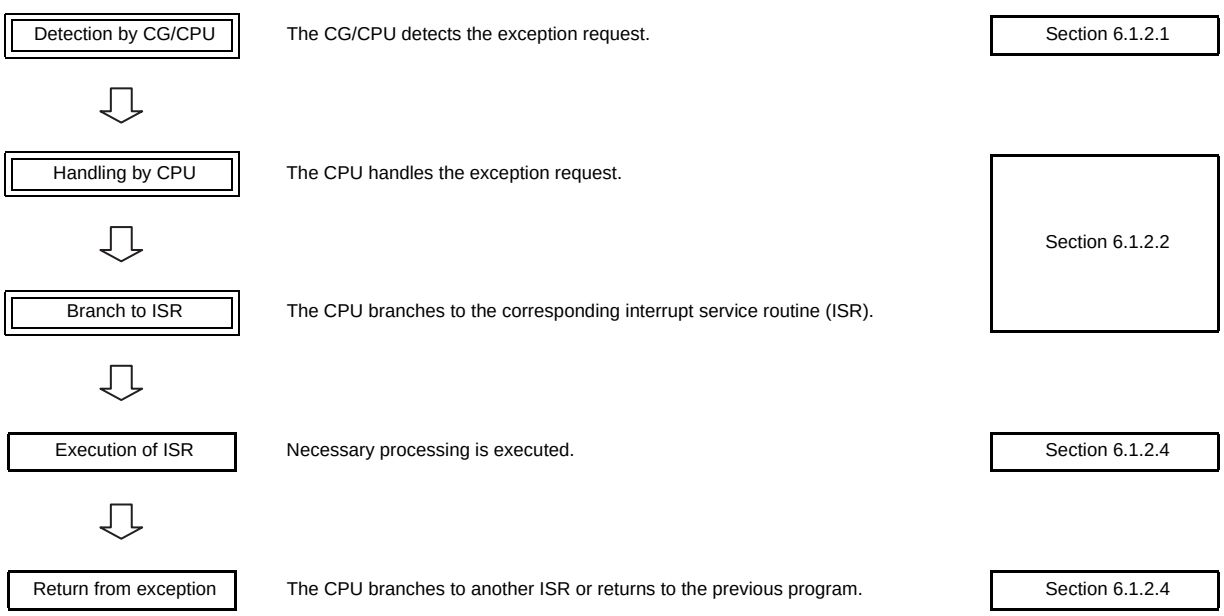
- Reset
- Non-Maskable Interrupt (NMI)
- Hard Fault
- Memory Management
- Bus Fault
- Usage Fault
- SVCall (Supervisor Call)
- Debug Monitor
- PendSV
- SysTick
- External Interrupt

6.1.2 Handling Flowchart

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

Each step is described later in this chapter.





6.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

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

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

An interrupt request is generated from an external interrupt pin or peripheral function.For interrupts that are used for releasing a standby mode, relevant settings must be made in the clock generator. For details, refer to "6.5 Interrupts".

(2) Exception detection

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

Table 6-1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 6-1 Exception Types and Priority

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

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

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

(3) Priority setting

- Priority level

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

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

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

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

- Priority grouping

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

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

The Table 6-2 shows the priority group setting. The pre-emption priority and the sub priority in the table are the number in the case that <PRI_n> is defined as an 8-bit configuration.

Table 6-2 Priority grouping setting

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

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

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

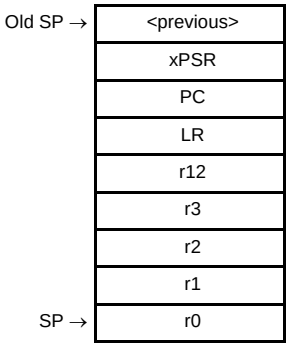
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

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

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

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



(2) fetching an ISR

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

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

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

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

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

6.1.2.3 Executing an ISR

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

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

For details about interrupt handling, see "6.5 Interrupts".

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

6.1.2.4 Exception exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

(2) Exception exit sequence

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

- Pop eight registers

  Pops the eight registers (PC, xPSR, r0 to r3, r12 and LR) from the stack and adjust the SP.
- Load current active interrupt number

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

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

6.2 Reset Exceptions

Reset exceptions are generated from the following six sources.

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

- External reset pin

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

- Reset exception by POR

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

- Reset exception by VLTD

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

- Reset exception by OFD

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

- Reset exception by WDT

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

- Reset exception by SYSRESETREQ

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

6.3 Non-Maskable Interrupts (NMI)

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

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

6.4 SysTick

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

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

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

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

6.5 Interrupts

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

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

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

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

6.5.1 Interrupt Sources

6.5.1.1 Interrupt route

Figure 6-1 shows an interrupt request route.

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

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

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

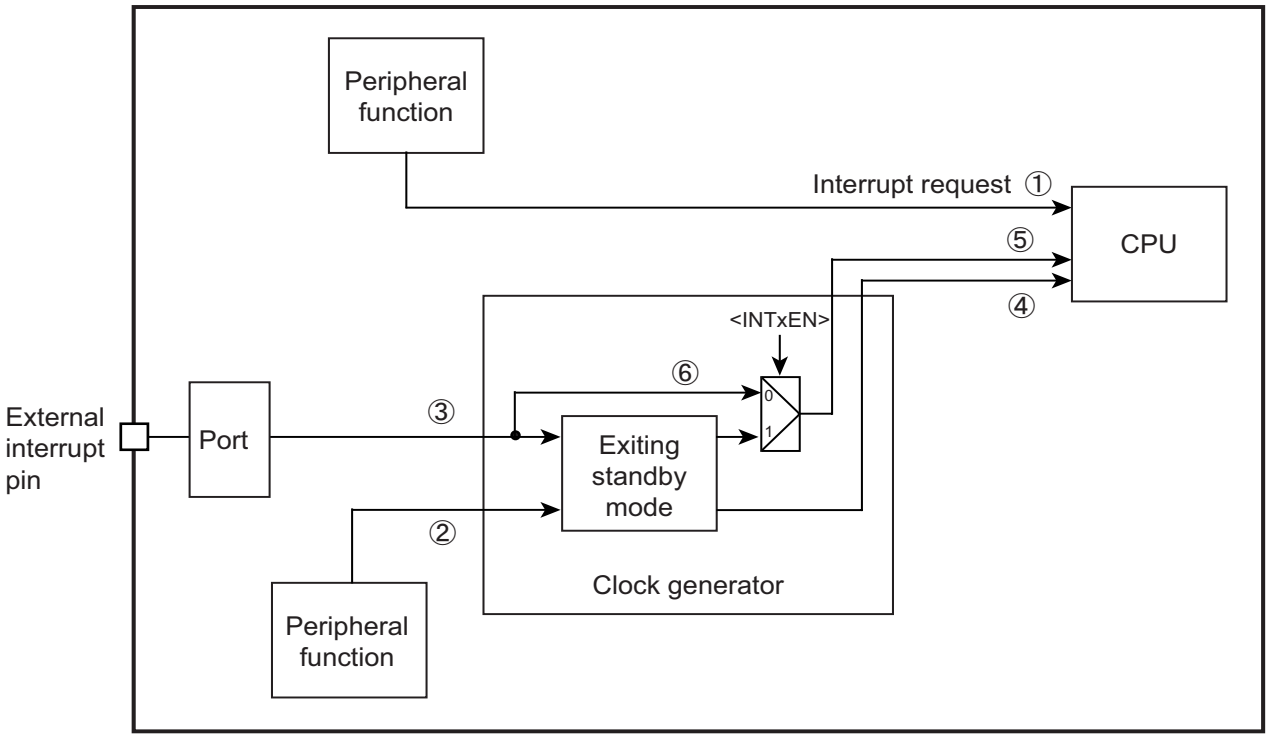


Figure 6-1 Interrupt Route

6.5.1.2 Generation

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

- From external pin

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

- From peripheral function

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

  See the chapter of each peripheral function for details.

- By setting Interrupt Set-Pending Register (forced pending)

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

6.5.1.3 Transmission

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

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

6.5.1.4 Precautions when using external interrupt pins

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

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

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

6.5.1.5 List of Interrupt Sources

Table 6-3 shows the list of interrupt sources.

Table 6-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGA
1	INT1	Interrupt Pin		
2	INT2	Interrupt Pin		
3	INT3	Interrupt Pin		
4	INT4	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGB
5	INT5	Interrupt Pin		
6	INTRX0	Serial reception (channel0)		
7	INTTX0	Serial transmit (channel0)		
8	INTRX1	Serial reception (channel1)		
9	INTTX1	Serial transmit (channel1)		
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		

Table 6-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
37	INTCAP51	16bit TMRB5 input capture 1		
38	INT6	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGB
39	INT7	Interrupt Pin		
40	INTRX2	Serial reception (channel2)		
41	INTTX2	Serial transmit (channel2)		
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	INTADBCPB	ADCB conversion monitoring function interrupt B		
46	INTTB20	16bit TMRB2 compare match detection 0/ Over flow		
47	INTTB21	16bit TMRB2 compare match detection 1		
48	INTTB30	16bit TMRB3 compare match detection 0/ Over flow		
49	INTTB31	16bit TMRB3 compare match detection 1		
50	INTCAP20	16bit TMRB2 input capture 0		
51	INTCAP21	16bit TMRB2 input capture 1		
52	INTCAP30	16bit TMRB3 input capture 0		
53	INTCAP31	16bit TMRB3 input capture 1		
54	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		

Table 6-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing standby)	CG interrupt mode control register
58	INT8	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGC
59	INT9	Interrupt Pin		
60	INTA	Interrupt Pin		
61	INTB	Interrupt Pin		
62	INTENC0	Encoder input0 interrupt		
63	INTENC1	Encoder input1 interrupt		
64	INTRX3	Serial reception (channel3)		
65	INTTX3	Serial transmit (channel3)		
66	INTTB60	16bit TMRB6 compare match detection 0/ Over flow		
67	INTTB61	16bit TMRB6 compare match detection 1		
68	INTTB70	16bit TMRB7 compare match detection 0/ Over flow		
69	INTTB71	16bit TMRB7 compare match detection 1		
70	INTCAP60	16bit TMRB6 input capture 0		
71	INTCAP61	16bit TMRB6 input capture 1		
72	INTCAP70	16bit TMRB7 input capture 0		
73	INTCAP71	16bit TMRB7 input capture 1		
74	INTC	Interrupt Pin	High/Low Edge/Level Selectable	CGIMCGD
75	INTD	Interrupt Pin		
76	INTE	Interrupt Pin		
77	INTF	Interrupt Pin		
78	INTSBI	Serial bus interface		

6.5.1.6 Active level

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

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

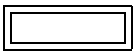
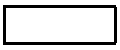
If an interrupt source is used for clearing a standby mode, setting the relevant clock generator register is also required. Enable the CGIMCGx<INTxEN> bit and specify the active level in the CGIMCGx<EMCGx> bits. You must set the active level for interrupt requests from each peripheral function as shown in Table 6-3

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

6.5.2 Interrupt Handling

6.5.2.1 Flowchart

The following shows how an interrupt is handled.

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

Processing	Details	See
Setting for detection	Set the relevant NVIC registers for detecting interrupts. Set the clock generator as well if each interrupt source is used to clear a standby mode. o Common setting NVIC registers o setting to clear standby mode Clock generator	"6.5.2.2 Preparation"
setting for sending interrupt signal	Execute an appropriate setting to send the interrupt signal depending on the interrupt type. o Setting for interrupt from external pin Port o Setting for interrupt from peripheral function Peripheral function (See the chapter of each peripheral function for details.)	
⇩		
Interrupt generation	An interrupt request is generated.	
Not clearing standby mode / Clearing standby mode CG detects interrupt (clearing standby mode)	Interrupt lines used for clearing a standby mode are connected to the CPU via the clock generator.	"6.5.2.3 Detection by Clock Generator"
CPU detects interrupt.	The CPU detects the interrupt. If multiple interrupt requests occur simultaneously, the interrupt request with the highest priority is detected according to the priority order.	"6.5.2.4 Detection by CPU"
⇩		
CPU handles interrupt.	The CPU handles the interrupt. The CPU pushes register contents to the stack before entering the ISR.	"6.5.2.5 CPU process-ing"
⇩		
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"6.5.2.6 Interrupt Service Routine (ISR)"
⇩		
Return to preceding program	Configure to return to the preceding program of the ISR.	

6.5.2.2 Preparation

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

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

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

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

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

(1) Disabling interrupt by CPU

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

Use "MSR" instruction to set this register.

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

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

(2) CPU registers setting

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note: m: corresponding bit

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "6.6.3.5 CGICRCG (CG Interrupt Request Clear Register)" for each value.

Interrupt requests from external pins can be used without setting the clock generator if they are not used for exiting a standby mode. However, an "High" pulse or "High"-level signal must be input so that the CPU can detect it as an interrupt request. Also, be aware of the description of "6.5.1.4 Precautions when using external interrupt pins".

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

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

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

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

6.5.2.3 Detection by Clock Generator

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

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

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

6.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

6.5.2.5 CPU processing

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

6.5.2.6 Interrupt Service Routine (ISR)

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

(1) Pushing during ISR

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

Push the contents of other registers if needed.

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

(2) Clearing an interrupt source

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

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

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

6.6 Exception / Interrupt-Related Registers

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

6.6.1 Register List

NVIC registers — Base Address = 0xE000_E000

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

Clock generator register Base Address = 0x4004_0200

Register name		Address
CG Interrupt Request Clear Register	CGICRCG	0x0014
NMI Flag Register	CGNMIFLG	0x0018
Reset Flag Register	CGRSTFLG	0x001C
CG Interrupt Mode Control Register A	CGIMCGA	0x0020
CG Interrupt Mode Control Register B	CGIMCGB	0x0024
CG Interrupt Mode Control Register C	CGIMCGC	0x0028
CG Interrupt Mode Control Register D	CGIMCGD	0x002C
Reserved	-	0x0030
Reserved	-	0x0034
Reserved	-	0x0038
Reserved	-	0x003C

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

6.6.2 NVIC Registers

6.6.2.1 SysTick Control and Status Register

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

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

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

6.6.2.2 SysTick Reload Value Register

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

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

6.6.2.3 SysTick Correct Value Register

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

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

6.6.2.4 SysTick Calibration Value Register

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

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

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

6.6.2.5 Interrupt Set-Enable Register 1

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

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

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

6.6.2.6 Interrupt Set-Enable Register 2

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

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

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

6.6.2.7 Interrupt Set-Enable Register 3

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

Bit	Bit Symbol	Type	Function
31-15	–	R/W	Read as 0.
14-0	SETENA	R/W	Interrupt number [78:64] [Write] 1: Enable [Read] 0: Disabled 1: Enabled Each bit corresponds to the specified number of interrupts. Writing "1" to a bit in this register enables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

6.6.2.8 Interrupt Clear-Enable Register 1

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

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

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

6.6.2.9 Interrupt Clear-Enable Register 2

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

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

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

6.6.2.10 Interrupt Clear-Enable Register 3

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

Bit	Bit Symbol	Type	Function
31-15	–	R/W	Read as 0.
14-0	CLRENA	R/W	Interrupt number [78:64] [Write] 1: Disabled [Read] 0: Disabled 1: Enable Each bit corresponds to the specified number of interrupts. It can be performed to enable interrupts and to check if interrupts are disabled. Writing "1" to a bit in this register disables the corresponding interrupt. Writing "0" has no effect. Reading the bits can see the enable/disable condition of the corresponding interrupts.

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

6.6.2.11 Interrupt Set-Pending Register 1

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

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

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

6.6.2.12 Interrupt Set-Pending Register 2

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

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

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

6.6.2.13 Interrupt Set-Pending Register 3

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

Bit	Bit Symbol	Type	Function
31-15	–	R/W	Read as 0.
14-0	SETPEND	R/W	Interrupt number [78:64] [Write] 1: Pend [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts. Writing "1" to a corresponding bit in the Interrupt Clear-Pending Register clears the bit in this register.

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

6.6.2.14 Interrupt Clear-Pending Register 1

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

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

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

6.6.2.15 Interrupt Clear-Pending Register 2

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

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

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

6.6.2.16 Interrupt Clear-Pending Register 3

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

Bit	Bit Symbol	Type	Function
31-15	–	R/W	Read as 0.
14-0	CLRPEND	R/W	Interrupt number [78:64] [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending Each bit corresponds to the specified number can force interrupts into the pending state and determines which interrupts are currently pending. Writing "1" to a bit in this register clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced. Writing "0" has no effect. Reading the bit returns the current state of the corresponding interrupts.

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

6.6.2.17 Interrupt Priority Register

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

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

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

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

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

6.6.2.18 Vector Table Offset Register

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

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

6.6.2.19 Application Interrupt and Reset Control Register

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

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

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

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

6.6.2.20 System Handler Priority Register

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

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

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

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

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

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

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

6.6.2.21 System Handler Control and State Register

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

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

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

6.6.3 Clock generator registers

6.6.3.1 CGIMCGA (CG Interrupt Mode Control Register A)

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

Bit	Bit Symbol	Type	Function
31	–	R	Read as 0.
30-28	EMCG3[2:0]	R/W	active level setting of INT3 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
27-26	EMST3[1:0]	R	active level of INT3 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
25	–	R	Reads as undefined.
24	INT3EN	R/W	INT3 clear input 0: Disable 1: Enable
23	–	R	Read as 0.
22-20	EMCG2[2:0]	R/W	active level setting of INT2 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
19-18	EMST2[1:0]	R	active level of INT2 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
17	–	R	Reads as undefined.
16	INT2EN	R/W	INT2 clear input 0:Disable 1: Enable
15	–	R	Read as 0.
14-12	EMCG1[2:0]	R/W	active level setting of INT1 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
11-10	EMST1[1:0]	R	active level of INT1 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
9	–	R	Reads as undefined.
8	INT1EN	R/W	INT1 clear input 0: Disable 1: Enable
7	–	R	Read as 0,
6-4	EMCG0[2:0]	R/W	active level setting of INT0 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
3-2	EMST0[1:0]	R	active level of INT0 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
1	–	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
0	INT0EN	R/W	INT0 clear input 0: Disable 1: Enable

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

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

6.6.3.2 CGIMCGB (CG Interrupt Mode Control Register B)

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

Bit	Bit Symbol	Type	Function
31	–	R	Read as 0.
30-28	EMCG7[2:0]	R/W	active level setting of INT7 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
27-26	EMST7[1:0]	R	active level of INT7 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
25	–	R	Reads as undefined.
24	INT7EN	R/W	INT7 clear input 0: Disable 1: Enable
23	–	R	Read as 0.
22-20	EMCG6[2:0]	R/W	active level setting of INT6 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
19-18	EMST6[1:0]	R	active level of INT6 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
17	–	R	Reads as undefined.
16	INT6EN	R/W	INT6 clear input 0:Disable 1: Enable
15	–	R	Read as 0.
14-12	EMCG5[2:0]	R/W	active level setting of INT5 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
11-10	EMST5[1:0]	R	active level of INT5 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
9	–	R	Reads as undefined.
8	INT5EN	R/W	INT5 clear input 0: Disable 1: Enable
7	–	R	Read as 0.
6-4	EMCG4[2:0]	R/W	active level setting of INT4 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
3-2	EMST4[1:0]	R	active level of INT4 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
1	–	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
0	INT4EN	R/W	INT4 clear input 0: Disable 1: Enable

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

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

6.6.3.3 CGIMCGC (CG Interrupt Mode Control Register C)

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

Bit	Bit Symbol	Type	Function
31	–	R	Read as 0.
30-28	EMCGB[2:0]	R/W	active level setting of INTB standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
27-26	EMSTB[1:0]	R	active level of INTB standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
25	–	R	Reads as undefined.
24	INTBEN	R/W	INTB clear input 0: Disable 1: Enable
23	–	R	Read as 0.
22-20	EMCGA[2:0]	R/W	active level setting of INTA standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
19-18	EMSTA[1:0]	R	active level of INTA standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
17	–	R	Reads as undefined.
16	INTAEN	R/W	INTA clear input 0:Disable 1: Enable
15	–	R	Read as 0.
14-12	EMCG9[2:0]	R/W	active level setting of INT9 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
11-10	EMST9[1:0]	R	active level of INT9 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
9	–	R	Reads as undefined.
8	INT9EN	R/W	INT9 clear input 0: Disable 1: Enable
7	–	R	Read as 0.
6-4	EMCG8[2:0]	R/W	active level setting of INT8 standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
3-2	EMST8[1:0]	R	active level of INT8 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
1	–	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
0	INT8EN	R/W	INT8 clear input 0: Disable 1: Enable

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

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

6.6.3.4 CGIMCGD (CG Interrupt Mode Control Register D)

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

Bit	Bit Symbol	Type	Function
31	–	R	Read as 0.
30-28	EMCGF[2:0]	R/W	active level setting of INTF standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
27-26	EMSTF[1:0]	R	active level of INTF standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
25	–	R	Reads as undefined.
24	INTFEN	R/W	INTF clear input 0: Disable 1: Enable
23	–	R	Read as 0.
22-20	EMCGE[2:0]	R/W	active level setting of INTE standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
19-18	EMSTE[1:0]	R	active level of INTE standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
17	–	R	Reads as undefined.
16	INTEEN	R/W	INTE clear input 0: Disable 1: Enable
15	–	R	Read as 0.
14-12	EMCGD[2:0]	R/W	active level setting of INTD standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
11-10	EMSTD[1:0]	R	active level of INTD standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
9	–	R	Reads as undefined.
8	INTDEN	R/W	INTD clear input 0: Disable 1: Enable
7	–	R	Read as 0.
6-4	EMCGC[2:0]	R/W	active level setting of INTC standby clear request. (101 to 111: setting prohibited) 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edge
3-2	EMSTC[1:0]	R	active level of INTC standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edge
1	–	R	Reads as undefined.

Bit	Bit Symbol	Type	Function
0	INTCEN	R/W	INTC clear input 0: Disable 1: Enable

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

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

6.6.3.5 CGICRCG (CG Interrupt Request Clear Register)

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

Bit	Bit Symbol	Type	Function
31-5	–	R	Read as 0,
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000:INT0 0_1000: INT8 0_0001: INT1 0_1001: INT9 0_0010: INT2 0_1010: INTA 0_0011: INT3 0_1011: INTB 0_0100: INT4 0_1100: INTC 0_0101: INT5 0_1101: INTD 0_0110: INT6 0_1110: INTE 0_ 0111: INT7 0_1111: INTF 1_0000 to 1_1111: Reserved Read as 0.

6.6.3.6 CGNMIFLG (NMI Flag Register)

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

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

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

6.6.3.7 CGRSTFLG (Reset Flag Register)

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

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

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

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

7. Input / Output Ports

7.1 Port Functions

7.1.1 Function list

TMPM376FDDFG/FDFG has 82 ports.Besides the ports function, these ports can be used as I/O pins for peripheral functions.

Table 7-1 shows the port function table.

Table 7-1 Port Function List

Port	PIn	Input /Out-put	Pull-up Pull-down	Schmitt Input	Noise Fil-ter	Program-mable Open-drain	Function pin
PORTA							
	PA0	I/O	Pull-up / Pull-down	o	o	o	TB0IN , INT3
	PA1	I/O	Pull-up / Pull-down	o	-	o	TB0OUT
	PA2	I/O	Pull-up / Pull-down	o	o	o	TB1IN , INT4
	PA3	I/O	Pull-up / Pull-down	o	-	o	TB1OUT
	PA4	I/O	Pull-up / Pull-down	o	-	o	SCLK1 , $\overline{\text{CTS1}}$
	PA5	I/O	Pull-up / Pull-down	o	-	o	TXD1 , TB6OUT
	PA6	I/O	Pull-up / Pull-down	o	-	o	RXD1 , TB6IN
	PA7	I/O	Pull-up / Pull-down	o	o	o	TB4IN , INT8
PORTB							
	PB0	I/O	Pull-up / Pull-down	o	-	o	TRACECLK
	PB1	I/O	Pull-up / Pull-down	o	-	o	TRACEDATA0
	PB2	I/O	Pull-up / Pull-down	o	-	o	TRACEDATA1
	PB3	I/O	Pull-up / Pull-down	o	-	o	TMS / SWDIO
	PB4	I/O	Pull-up / Pull-down	o	-	o	TCK / SWCLK
	PB5	I/O	Pull-up / Pull-down	o	-	o	TDO / SWV
	PB6	I/O	Pull-up / Pull-down	o	-	o	TDI
	PB7	I/O	Pull-up / Pull-down	o	o	o	$\overline{\text{TRST}}$
PORTC							
	PC0	I/O	Pull-up / Pull-down	o	-	o	UO0
	PC1	I/O	Pull-up / Pull-down	o	-	o	XO0
	PC2	I/O	Pull-up / Pull-down	o	-	o	VO0
	PC3	I/O	Pull-up / Pull-down	o	-	o	YO0
	PC4	I/O	Pull-up / Pull-down	o	-	o	WO0
	PC5	I/O	Pull-up / Pull-down	o	-	o	ZO0
	PC6	I/O	Pull-up / Pull-down	o	-	o	$\overline{\text{EMG0}}$
	PC7	I/O	Pull-up / Pull-down	o	-	o	$\overline{\text{OVV0}}$
PORTD							
	PD0	I/O	Pull-up / Pull-down	o	-	o	ENCA0 , TB5IN

o : Exist
- : Not Exist

Table 7-1 Port Function List

Port	PIn	Input /Output	Pull-up Pull-down	Schmitt Input	Noise Filter	Programmable Open-drain	Function pin
	PD1	I/O	Pull-up / Pull-down	o	-	o	ENCB0 , TB5OUT
	PD2	I/O	Pull-up / Pull-down	o	-	o	ENCZ0
	PD3	I/O	Pull-up / Pull-down	o	o	o	INT9
	PD4	I/O	Pull-up / Pull-down	o	-	o	SCLK2 , $\overline{\text{CTS2}}$
	PD5	I/O	Pull-up / Pull-down	o	-	o	TXD2
	PD6	I/O	Pull-up / Pull-down	o	-	o	RXD2
PORTE							
	PE0	I/O	Pull-up / Pull-down	o	-	o	TXD0
	PE1	I/O	Pull-up / Pull-down	o	-	o	RXD0
	PE2	I/O	Pull-up / Pull-down	o	-	o	SCLK0 , $\overline{\text{CTS0}}$
	PE3	I/O	Pull-up / Pull-down	o	-	o	TB4OUT
	PE4	I/O	Pull-up / Pull-down	o	o	o	TB2IN , INT5
	PE5	I/O	Pull-up / Pull-down	o	-	o	TB2OUT
	PE6	I/O	Pull-up / Pull-down	o	o	o	TB3IN , INT6
	PE7	I/O	Pull-up / Pull-down	o	o	o	TB3OUT , INT7
PORTF							
	PF0	I/O	Pull-up / Pull-down	o	-	o	TB7IN , $\overline{\text{BOOT}}$
	PF1	I/O	Pull-up / Pull-down	o	-	o	TB7OUT
	PF2	I/O	Pull-up / Pull-down	o	-	o	ENCA1 , SCLK3 , $\overline{\text{CTS3}}$
	PF3	I/O	Pull-up / Pull-down	o	-	o	ENCB1 , TXD3
	PF4	I/O	Pull-up / Pull-down	o	-	o	ENCZ1 , RXD3
PORTG							
	PG0	I/O	Pull-up / Pull-down	o	-	o	UO1
	PG1	I/O	Pull-up / Pull-down	o	-	o	XO1
	PG2	I/O	Pull-up / Pull-down	o	-	o	VO1
	PG3	I/O	Pull-up / Pull-down	o	-	o	YO1
	PG4	I/O	Pull-up / Pull-down	o	-	o	WO1
	PG5	I/O	Pull-up / Pull-down	o	-	o	ZO1
	PG6	I/O	Pull-up / Pull-down	o	-	o	$\overline{\text{EMG1}}$
	PG7	I/O	Pull-up / Pull-down	o	-	o	$\overline{\text{OVV1}}$
PORTH							
	PH0	I/O	Pull-up / Pull-down	o	o	o	INT0 , AINA0
	PH1	I/O	Pull-up / Pull-down	o	o	o	INT1 , AINA1
	PH2	I/O	Pull-up / Pull-down	o	o	o	INT2 , AINA2
	PH3	I/O	Pull-up / Pull-down	o	-	o	AINA3
	PH4	I/O	Pull-up / Pull-down	o	-	o	AINA4
	PH5	I/O	Pull-up / Pull-down	o	-	o	AINA5
	PH6	I/O	Pull-up / Pull-down	o	-	o	AINA6
	PH7	I/O	Pull-up / Pull-down	o	-	o	AINA7
PORTI							

o : Exist
- : Not Exist

Table 7-1 Port Function List

Port	PIn	Input /Output	Pull-up Pull-down	Schmitt Input	Noise Filter	Programmable Open-drain	Function pin
	PI0	I/O	Pull-up / Pull-down	o	-	o	AINA8
	PI1	I/O	Pull-up / Pull-down	o	-	o	AINA9 / AINB0
	PI2	I/O	Pull-up / Pull-down	o	-	o	AINA10 / AINB1
	PI3	I/O	Pull-up / Pull-down	o	-	o	AINA11 / AINB2
PORTJ							
	PJ0	I/O	Pull-up / Pull-down	o	-	o	AINB3
	PJ1	I/O	Pull-up / Pull-down	o	-	o	AINB4
	PJ2	I/O	Pull-up / Pull-down	o	-	o	AINB5
	PJ3	I/O	Pull-up / Pull-down	o	-	o	AINB6
	PJ4	I/O	Pull-up / Pull-down	o	-	o	AINB7
	PJ5	I/O	Pull-up / Pull-down	o	-	o	AINB8
	PJ6	I/O	Pull-up / Pull-down	o	o	o	INTC , AINB9
	PJ7	I/O	Pull-up / Pull-down	o	o	o	INTD , AINB10
PORTK							
	PK0	I/O	Pull-up / Pull-down	o	o	o	INTE , AINB11
	PK1	I/O	Pull-up / Pull-down	o	o	o	INTF , AINB12
PORTL							
	PL0	Input	-	o	o	-	INTB
	PL1	Input	-	o	o	-	INTA
PORTM							
	PM0	I/O	Pull-up / Pull-down	o	-	o	X1
	PM1	I/O	Pull-up / Pull-down	o	-	o	X2
PORTN							
	PN0	I/O	Pull-up / Pull-down	o	-	o	SO / SDA
	PN1	I/O	Pull-up / Pull-down	o	-	o	SI / SCL
	PN2	I/O	Pull-up / Pull-down	o	-	o	SCK
	PN3	I/O	Pull-up / Pull-down	o	-	o	TB7IN

o : Exist
- : Not Exist

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

7.1.2 Port Registers Outline

The following registers need to be configured to use ports.

- PxDATA: Port x data register

  To read / write port data.

- PxCR: Port x output control register

  To control output.

  PxIE needs to be configured to control input.

- PxFRn: Port x function register n

  To set function.

  An assigned function can be activated by setting "1".

- PxOD: Port x open drain control register

  To control the programmable open drain.

  Programmable open drain is function to be materialized pseudo-open-drain by setting the PxOD.

  When PxOD is set "1",output buffer is disabled and pseudo-open-drain is materialized.

- PxPUP: Port x pull-up control register

  To control programmable pull ups.

- PxPDN: Port x pull-down control register

  To control programmable pull downs.

- PxIE:Port x input control register

  To control inputs.

  For avoided through current, default setting prohibits inputs.

7.1.3 Port States in STOP Mode

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

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

Table 7-2 shows the pin conditions in STOP mode.

Table 7-2 Port conditions in STOP mode

	Pin name	I/O	<DRVE> = 0	<DRVE> = 1
Not port	$\overline{\text{RESET}}$, MODE	Input only	o	
Port	X1	Input only	×	
	X2	Output only	"High" level output	
	TMS TCK TDI TRST	Input	o	
	TDO	Output	Enabled when data is valid. Disabled when data is invalid.	
	SWCLK	Input	o	
	SWDIO	Input	o	
		Output	Enabled when data is valid. Disabled when data is invalid.	
	TRACECLK TRACEDATA0 TRACEDATA1 SWV	Output	o	
	UO0,1 VO0,1 WO0,1 XO0,1 YO0,1 ZO0,1	Output	Enabled when data is valid. Disabled when data is invalid.	
	INT0, INT1, INT2 INT3, INT4, INT5 INT6, INT7, INT8 INT9, INTA, INTB INTC, INTD, INTE INTF	Input	o	
	Other function pins other than the above or the ports that are used as general purpose ports.	Input	×	o
		Output	×	o

o : Input or output enabled.
× : Input or output disabled.

7.2 Port functions

This chapter describes the port registers detail.

This chapter describes only "circuit type" reading circuit configuration. For detailed circuit diagram, refer to"7.3 Block Diagrams of Ports".

7.2.1 Port A (PA0 to PA7)

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

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

The Port A has two types of function register. If you use the port A as a general-purpose port, set "0" to the corresponding bit of the two registers.If you use the port A as other than a general-purpose port, set "1" to the corresponding bit of the function register.Do not set "1" to the some function registers at the same time.

To use the external interrupt input for releasing STOP mode, select this function in the PAFR 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 if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

7.2.1.1 Port A Circuit Type

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

7.2.1.2 PortA register

Base Address = 0x4000_0000

Register name		Address (Base+)
Port A data register	PADATA	0x0000
Port A output control register	PACR	0x0004
Port A function register 1	PAFR1	0x0008
Port A function register2	PAFR2	0x000C
Port A open drain control register	PAOD	0x0028
Port A pull-up control register	PAPUP	0x002C
Port A pull-down control register	PAPDN	0x0030
Port A input control register	PAIE	0x0038

7.2.1.3 PADATA (Port A data register)

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

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

7.2.1.4 PACR (Port A output control register)

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

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

7.2.1.5 PAFR1 (Port A function register 1)

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

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

7.2.1.6 PAFR2 (Port A function register 2)

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

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

7.2.1.7 PAOD (Port A open drain control register)

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

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

7.2.1.8 PAPUP (Port A pull-up control register)

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

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

7.2.1.9 PAPDN (Port A pull-down control register)

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

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

7.2.1.10 PAIE (Port A input control register)

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

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

7.2.2 Port B (PB0 to PB7)

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

Reset initializes PB3, PB4, PB5, PB6 and PB7 to perform debug interface function.

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

When PB5 functions as the TDO or SWV, output is enabled. When PB6 functions TDI, input, pull-up are enabled. When PB7 functions as $\overline{\text{TRST}}$ input, pull-up is enabled.

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

Note: If PB3 is configured as the TMS/SWDIO pin, output is enabled even in STOP mode regardless of the CGSTBYCR<DRVE> bit setting.

7.2.2.1 Port B Circuit Type

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

7.2.2.2 Port B register

Base Address = 0x4000_0040

Register name		Address(Base+)
Port B data register	PBDATA	0x0000
Port B output control register	PBCR	0x0004
Port B function register 1	PBFR1	0x0008
Port B open drain control register	PBOD	0x0028
Port B pull-up control register	PBPUP	0x002C
Port B pull-down control register	PBPDN	0x0030
Port B input control register	PBIE	0x0038

7.2.2.3 PBDATA (Port B data register)

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

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

7.2.2.4 PBCR (Port B output control register)

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

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

7.2.2.5 PBFR1 (Port B function register 1)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	PB7F1	R/W	0 : PORT 1 : $\overline{\text{TRST}}$
6	PB6F1	R/W	0: PORT 1: TDI
5	PB5F1	R/W	0: PORT 1: TDO / SWV
4	PB4F1	R/W	0: PORT 1: TCK / SWCLK
3	PB3F1	R/W	0: PORT 1: TMS / SWDIO
2	PB2F1	R/W	0: PORT 1: TRACEDATA1
1	PB1F1	R/W	0: PORT 1: TRACEDATA0
0	PB0F1	R/W	0: PORT 1: TRACECLK

7.2.2.6 PBOD (Port B open drain control register)

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

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

7.2.2.7 PBPUP (Port B pull-up control register)

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

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

7.2.2.8 PBPDN (Port B pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PB7DN	PB6DN	PB5DN	PB4DN	PB3DN	PB2DN	PB1DN	PB0DN
After reset	0	0	0	1	0	0	0	0

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

7.2.2.9 PBIE (Port B input control register)

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

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

7.2.3 Port C (PC0 to PC7)

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

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

7.2.3.1 Port C Circuit Type

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

7.2.3.2 Port C register

Base Address = 0x4000_0080

Register name		Address(Base+)
Port C data register	PCDATA	0x0000
Port C output control register	PCCR	0x0004
Port C function register 1	PCFR1	0x0008
Port C open drain control register	PCOD	0x0028
Port C pull-up control register	PCPUP	0x002C
Port C pull-down control register	PCPDN	0x0030
Port C input control register	PCIE	0x0038

7.2.3.3 PCDATA (Port C data register)

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

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

7.2.3.4 PCCR (Port C output control register)

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

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

7.2.3.5 PCFR1 (Port C function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7F1	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	PC7F1	R/W	0 : PORT 1 : $\overline{OVV0}$
6	PC6F1	R/W	0: PORT 1:$\overline{EMG0}$
5	PC5F1	R/W	0: PORT 1: ZO0
4	PC4F1	R/W	0: PORT 1: WO0
3	PC3F1	R/W	0: PORT 1: YO0
2	PC2F1	R/W	0: PORT 1: VO0
1	PC1F1	R/W	0: PORT 1: XO0
0	PC0F1	R/W	0: PORT 1: UO0

7.2.3.6 PCOD (Port C open drain control register)

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

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

7.2.3.7 PCPUP (Port C pull-up control register)

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

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

7.2.3.8 PCPDN (Port C pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PC7DN	PC6DN	PC5DN	PC4DN	PC3DN	PC2DN	PC1DN	PC0DN
After reset	0	0	0	0	0	0	0	0

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

7.2.3.9 PCIE (Port C input control register)

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

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

7.2.4 Port D (PD0 to PD6)

The port D is a general-purpose, 7-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port D performs the serial interface function (SIO / UART), the external signal interrupt input, the 16-bit timer input/output function and the Encoder input function.

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

The Port D has two types of function register. If you use the port D as a general-purpose port, set "0" to the corresponding bit of the two registers.If you use the port D as other than a general-purpose port, set "1" to the corresponding bit of the function register.Do not set "1" to the some function registers at the same time.

To use the external interrupt input for releasing STOP mode, select this function in the PDFR1 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 if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

7.2.4.1 Port D Circuit Type

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

7.2.4.2 Port D register

Base Address = 0x4000_00C0

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

7.2.4.3 PDDATA (Port D data register)

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

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

7.2.4.4 PDCR (Port D output control register)

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

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

7.2.4.5 PDFR1 (Port D function register 1)

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

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

7.2.4.6 PDFR2 (Port D function register 2)

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

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

7.2.4.7 PDOD (Port D open drain control register)

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

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

7.2.4.8 PDPUP (Port D pull-up control register)

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

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

7.2.4.9 PDPDN (Port D pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	PD6DN	PD5DN	PD4DN	PD3DN	PD2DN	PD1DN	PD0DN
After reset	0	0	0	0	0	0	0	0

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

7.2.4.10 PDIE (Port D input control register)

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

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

7.2.5 Port E (PE0 to PE7)

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

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

The Port E has two types of function register. If you use the port E as a general-purpose port, set "0" to the corresponding bit of the two registers.If you use the port E as other than a general-purpose port, set "1" to the corresponding bit of the function register.Do not set "1" to the some function registers at the same time.

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

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

7.2.5.1 Port E Circuit Type

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

7.2.5.2 Port E register

Base Address = 0x4000_0100

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

7.2.5.3 PEDATA (Port E data register)

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

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

7.2.5.4 PECR (Port E output control register)

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

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

7.2.5.5 PEFR1 (Port E function register 1)

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

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

7.2.5.6 PEFR2 (Port E function register 2)

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

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

7.2.5.7 PEOD (Port E open drain control register)

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

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

7.2.5.8 PEPUP (Port E pull-up control register)

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

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

7.2.5.9 PEPDN (Port E pull-down control register)

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

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

7.2.5.10 PEIE (Port E input control register)

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

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

7.2.6 Port F (PF0 to PF4)

The port F is a general-purpose, 5-bit input / output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input / output function, the port F performs the serial interface function (SIO / UART), the 16-bit timer input/output function, the Encoder input function 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, pull-up and pull-down disabled.

7.2.6.1 Port F Circuit Type

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

7.2.6.2 Port F register

Base Address = 0x4000_0140

Register name		Address(Base+)
Port F data register	PFDATA	0x0000
Port F output control register	PFCR	0x0004
Port F function register 1	PFFR1	0x0008
Port F function register 2	PFFR2	0x000C
Port F function register 3	PFFR3	0x0010
Port F open drain control register	PFOD	0x0028
Port F pull-up control register	PFPUP	0x002C
Port F pull-down control register	PFPDN	0x0030
Port F input control register	PFIE	0x0038

7.2.6.3 PFDATA (Port F data register)

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

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

7.2.6.4 PFCR (Port F output control register)

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

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

7.2.6.5 PFFR1 (Port F function register 1)

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

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

7.2.6.6 PFFR2 (Port F function register 2)

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

Bit	Bit Symbol	Type	Function
31-5	–	R	Read as 0.
4	PF4F2	R/W	0: PORT 1: RXD3
3	PF3F2	R/W	0: PORT 1: TXD3
2	PF2F2	R/W	0: PORT 1: SCLK3
1-0	–	R	Read as 0.

7.2.6.7 PFFR3 (Port F function register 3)

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

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

7.2.6.8 PFOD (Port F open drain control register)

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

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

7.2.6.9 PFPUP (Port F pull-up control register)

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

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

7.2.6.10 PFPDN (Port F pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PF4DN	PF3DN	PF2DN	PF1DN	PF0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	–	R	Read as 0.
4-0	PF4DN to PF0DN	R/W	Pull-down 0: Disable 1: Enable

7.2.6.11 PFIE (Port F input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
After reset	0	0	0	0	0	0	0	0

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

7.2.7 Port G (PG0 to PG7)

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

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

7.2.7.1 Port G Circuit Type

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

7.2.7.2 Port G register

Base Address = 0x4000_0180

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

7.2.7.3 PGDATA (Port G data register)

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

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

7.2.7.4 PGCR (Port G output control register)

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

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

7.2.7.5 PGFR1 (Port G function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PG7F1	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	PG7F1	R/W	0: PORT 1:$\overline{\text{OVV1}}$
6	PG6F1	R/W	0: PORT 1: $\overline{\text{EMG1}}$
5	PG5F1	R/W	0: PORT 1: ZO1
4	PG4F1	R/W	0: PORT 1: WO1
3	PG3F1	R/W	0: PORT 1: YO1
2	PG2F1	R/W	0: PORT 1: VO1
1	PG1F1	R/W	0: PORT 1: XO1
0	PG0F1	R/W	0: PORT 1: UO1

7.2.7.6 PGOD (Port G open drain control register)

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

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

7.2.7.7 PGPUP (Port G pull-up control register)

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

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

7.2.7.8 PGPDN (Port G pull-down control register)

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

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

7.2.7.9 PGIE (Port G input control register)

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

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

7.2.8 Port H (PH0 to PH7)

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

Note 2: In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

7.2.8.1 Port H Circuit Type

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

7.2.8.2 Port H register

Base Address = 0x4000_01C0

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

7.2.8.3 PHDATA (Port H data register)

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

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

7.2.8.4 PHCR (Port H output control register)

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

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

7.2.8.5 PHFR1 (Port H function register 1)

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

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

7.2.8.6 PHOD (Port H open drain control register)

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

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

7.2.8.7 PHPUP (Port H pull-up control register)

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

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

7.2.8.8 PHPDN (Port H pull-down control register)

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

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

7.2.8.9 PHIE (Port H input control register)

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

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

7.2.9 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 AD converter.

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

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

7.2.9.1 Port I Circuit Type

	7	6	5	4	3	2	1	0
Type	–	–	–	–	T16	T16-	T16	T16

7.2.9.2 Port I register

Base Address = 0x4000_0200

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

7.2.9.3 PIDATA (Port I data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3	PI2	PI1	PI0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PI3 to PI0	R/W	Port I data register

7.2.9.4 PICR (Port I output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3C	PI2C-	PI1C-	PI0C-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PI3C to PI0C	R/W	Output 0: Disable 1: Enable

7.2.9.5 PIOD (Port I open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3OD	PI2OD	PI1OD	PI0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PI3OD-PI0OD	R/W	0 : CMOS 1 : Open-drain

7.2.9.6 PIPUP (Port I pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3UP	PI2UP	PI1UP	PI0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PI3UP to PI0UP	R/W	Pull-up 0: Disable 1: Enable

7.2.9.7 PIPDN (Port I pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3DN	PI2DN	PI1DN	PI0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PI3DN to PI0DN	R/W	Pull-down 0: Disable 1: Enable

7.2.9.8 PIIE (Port I input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PI3IE	PI2IE	PI1IE	PI0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PI3IE to PI0IE	R/W	Input 0: Disable 1: Enable

7.2.10 Port J (PJ0 to PJ7)

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

Note 2: In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

7.2.10.1 Port J Circuit Type

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

7.2.10.2 Port J register

Base Address = 0x4000_0240

Register name		Address(Base+)
Port J data register	PJDATA	0x0000
Port J output control register	PJCR	0x0004
Port J function register 1	PJFR1	0x0008
Port J open drain control register	PJOD	0x0028
Port J pull-up control register	PJPUP	0x002C
Port J pull-down control register	PJPDN	0x0030
Port J input control register	PJIE	0x0038

7.2.10.3 PJDATA (Port J data register)

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

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

7.2.10.4 PJCR (Port J output control register)

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

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

7.2.10.5 PJFR1 (Port J function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7F1	PJ6F1	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	PJ7F1	R/W	0: PORT 1: INTD
6	PJ6F1	R/W	0: PORT 1: INTC
5-0	–	R	Read as 0.

7.2.10.6 PJOD (Port J open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7OD	PJ6OD	PJ5OD	PJ4OD	PJ3OD	PJ2OD	PJ1OD	PJ0OD
After reset	0	0	0	0	0	0	0	0

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

7.2.10.7 PJPUP (Port J pull-up control register)

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

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

7.2.10.8 PJPDN (Port J pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7DN	PJ6DN	PJ5DN	PJ4DN	PJ3DN	PJ2DN	PJ1DN	PJ0DN
After reset	0	0	0	0	0	0	0	0

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

7.2.10.9 PJIE (Port J input control register)

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

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

7.2.11 Port K (PK0 to PK1)

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

Note 2: In modes other than STOP mode, interrupt input is enabled regardless of the PxFR register setting if input is enabled in PxIE. Make sure to disable unused interrupts when programming the device.

7.2.11.1 Port K Circuit Type

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

7.2.11.2 Port K register

Base Address = 0x4000_0280

Register name		Address(Base+)
Port K data register	PKDATA	0x0000
Port K output control register	PKCR	0x0004
Port K function register 1	PKFR1	0x0008
Port K open drain control register	PKOD	0x0028
Port K pull-up control register	PKPUP	0x002C
Port K pull-down control register	PKPDN	0x0030
Port K input control register	PKIE	0x0038

7.2.11.3 PKDATA (Port K data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1	PK0
After reset	0	0	0	0	0	0	0	0

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

7.2.11.4 PKCR (Port K output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-				PK1C	PK0C
After reset	0	0	0	0	0	0	0	0

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

7.2.11.5 PKFR1 (Port K function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1F1	PK0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	PK1F1	R/W	0: PORT 1: INTF
0	PK0F1	R/W	0: PORT 1: INTE

7.2.11.6 PKOD (Port K open drain control register)

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

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

7.2.11.7 PKPUP (Port K pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1UP	PK0UP
After reset	0	0	0	0	0	0	0	0

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

7.2.11.8 PKPDN (Port K pull-down control register)

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

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

7.2.11.9 PKIE (Port K input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PK1IE	PK0IE
After reset	0	0	0	0	0	0	0	0

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

7.2.12 Port L (PL0 to PL1)

The port L 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 L performs the external signal interrupt input.

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

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

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

Note 2: When the power supply switch on, please keep 'Low' level to port L, constant time.(include in reset time) The details please watch 'Notice for the power supply' of 'Electrical Characteristics'.

7.2.12.1 Port L Circuit Type

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

7.2.12.2 Port L register

Base Address = 0x4000_02C0

Register name		Address(Base+)
Port L data register	PLDATA	0x0000
Port L function register 1	PLFR1	0x0008
Port L input control register	PLIE	0x0038

7.2.12.3 PLDATA (Port L data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PL1	PL0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1-0	PL1 to PL0	R/W	Port L data register

7.2.12.4 PLFR1 (Port L function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PL1F1	PL0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	PL1F1	R/W	0: PORT 1: INTA
0	PL0F1	R/W	0: PORT 1: INTB

7.2.12.5 PLIE (Port L input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PL1IE	PL0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1-0	PL1IE to PL0IE	R/W	Input 0: Disable 1: Enable

7.2.13 Port M (PM0 to PM1)

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

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

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

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

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

7.2.13.1 Port M Circuit Type

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

7.2.13.2 Port M register

Base Address = 0x4000_0300

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

7.2.13.3 PMDATA (Port M data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PM1	PM0
After reset	0	0	0	0	0	0	0	0

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

7.2.13.4 PMCR (Port M output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PM1C	PM0C
After reset	0	0	0	0	0	0	0	0

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

7.2.13.5 PMOD (Port M open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PM1OD	PM0OD
After reset	0	0	0	0	0	0	0	0

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

7.2.13.6 PMPUP (Port M pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PM1UP	PM0UP
After reset	0	0	0	0	0	0	0	0

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

7.2.13.7 PMPDN (Port M pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PM1DN	PM0DN
After reset	0	0	0	0	0	0	0	0

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

7.2.13.8 PMIE (Port M input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PM1IE	PM0IE
After reset	0	0	0	0	0	0	0	0

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

7.2.14 Port N (PN0 to PN3)

The port N 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 N performs the serial bus interface function (I2C / SIO) and the 16-bit timer input function.

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

7.2.14.1 Port N Circuit Type

	7	6	5	4	3	2	1	0
Type	-	-	-	-	T3	T22	T22	T22

7.2.14.2 Port N register

Base Address = 0x4000_0340

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

7.2.14.3 PNDATA (Port N data register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3	PN2	PN1	PN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PN3 to PN0	R/W	Port N data register

7.2.14.4 PNCR (Port N output control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3C	PN2C	PN1C	PN0C
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PN3C to PN0C	R/W	Output 0: Disable 1: Enable

7.2.14.5 PNFR1 (Port N function register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3F1	PN2F1	PN1F1	PN0F1
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3	PN3F1	R/W	0: PORT 1: TB7IN (Note)
2	PN2F1	R/W	0: PORT 1: SCK
1	PN1F1	R/W	0: PORT 1: SI / SCL
0	PN0F1	R/W	0: PORT 1: SO / SDA

Note: When you set 1 to <PN3F1> (selected TB7IN), please set 0 to <PF0F1>. However, the input of PF0 becomes effective when 1 is set to both <PN3F1> and <PF0F1>.

7.2.14.6 PNOD (Port N open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3OD	PN2OD	PN1OD	PN0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PN3OD to PN0OD	R/W	0 : CMOS 1 : Open-drain

7.2.14.7 PNPUP (Port N pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3UP	PN2UP	PN1UP	PN0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PN3UP to PN0UP	R/W	Pull-up 0: Disable 1: Enable

7.2.14.8 PNPDN (Port N pull-down control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3DN	PN2DN	PN1DN	PN0DN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PN3DN to PN0DN	R/W	Pull-down 0: Disable 1: Enable

7.2.14.9 PNIE (Port N input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	PN3IE	PN2IE	PN1IE	PN0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	PN3IE to PN0IE	R/W	Input 0: Disable 1: Enable

7.3 Block Diagrams of Ports

7.3.1 Port Types

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

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

Table 7-3 Function Lists

Type	GP Port	Function1	Function2	Function3	Analog	Pull-up	Pull-dn	Program-mable open-drain	Note
T1	I / O	Output	–	–	–	R	R	o	Function output triggered by enable signal
T2	I / O	Output	–	–	–	R	R	o	
T3	I / O	Input	–	–	–	R	R	o	
T4	I / O	Input (int)	–	–	–	R	R	o	
T5	Input	Input (int)	–	–	–	–	–	–	
T6	I / O	I / O	–	–	–	NoR	–	–	Function output triggered by enable signal
T7	I / O	Input	–	–	–	NoR	–	–	
T8	I / O	Input	–	–	–	–	NoR	–	
T9	I / O	I / O	Input	–	–	R	R	o	
T10	I / O	Input	Output	–	–	R	R	o	
T11	I / O	Input	Input	–	–	R	R	o	
T12	I / O	Input	Input(int)	–	–	R	R	o	
T13	I / O	Output	Output	–	–	R	R	o	
T14	I / O	Output	I / O	–	–	R	R	o	
T15	I / O	Input	I / O	Input	–	R	R	o	
T16	I / O	–	–	–	o	R	R	o	
T17	I / O	Input(int)	–	–	o	R	R	o	
T18	I / O	Output	–	–	–	R	–	–	
T19	I / O	Output	–	–	–	NoR	–	–	Function output triggered by enable signal
T20	I / O	Input	–	–	–	NoR	NoR	o	$\overline{\text{BOOT}}$ input enabled during reset
T21	I / O	- (OSC1)	–	–	–	R	R	o	High-speed osscillator (External)
T22	I / O	I / O	–	–	–	R			

int : Interrupt input
- : Not exist
o : Exist

R: Forced disable during reset
NoR: Unaffected by reset

7.3.2 Type T1

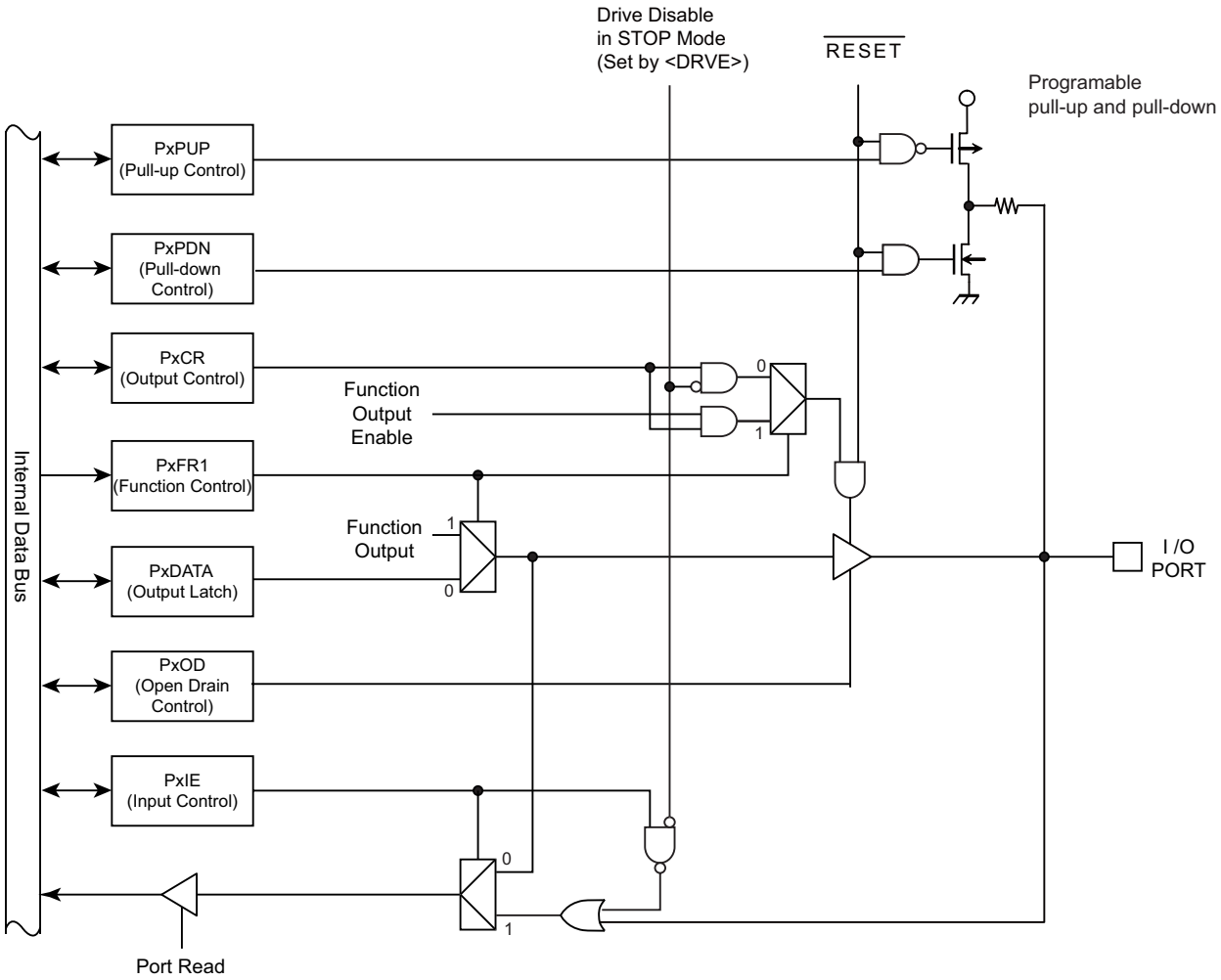


Figure 7-1 Port Type T1

7.3.3 Type T2

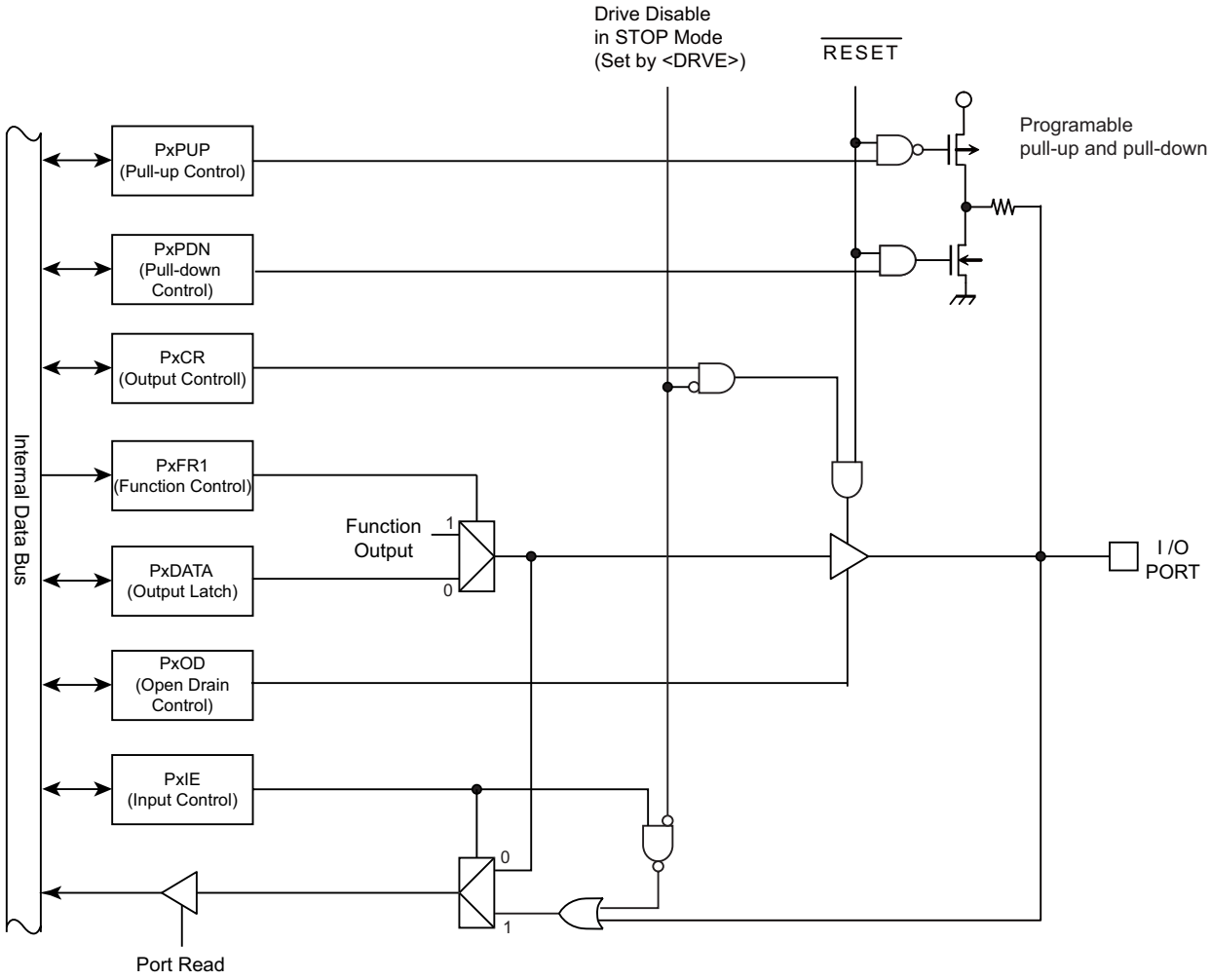


Figure 7-2 PORT Type T2

7.3.4 Type T3

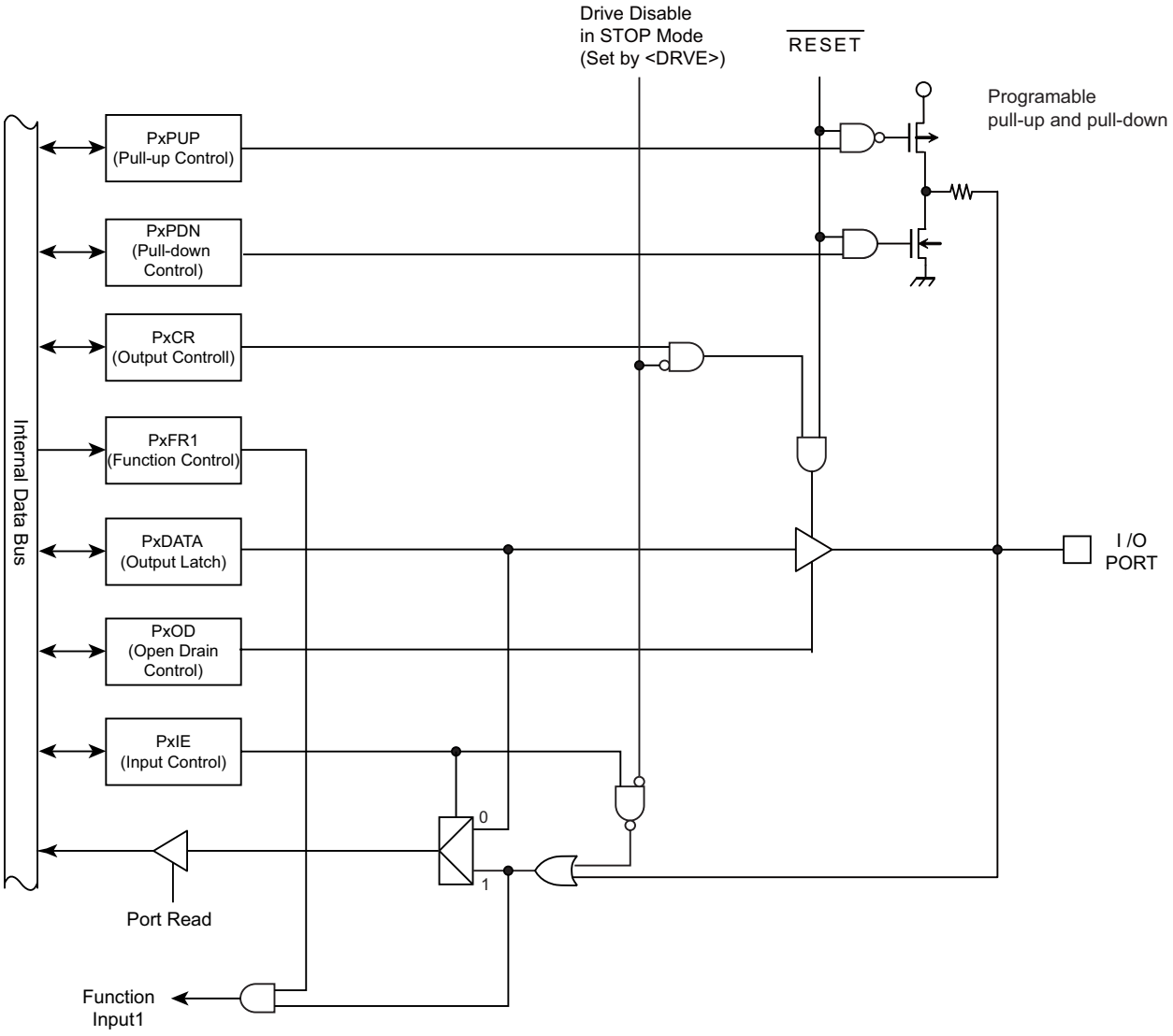


Figure 7-3 PORT Type T3

7.3.5 Type T4

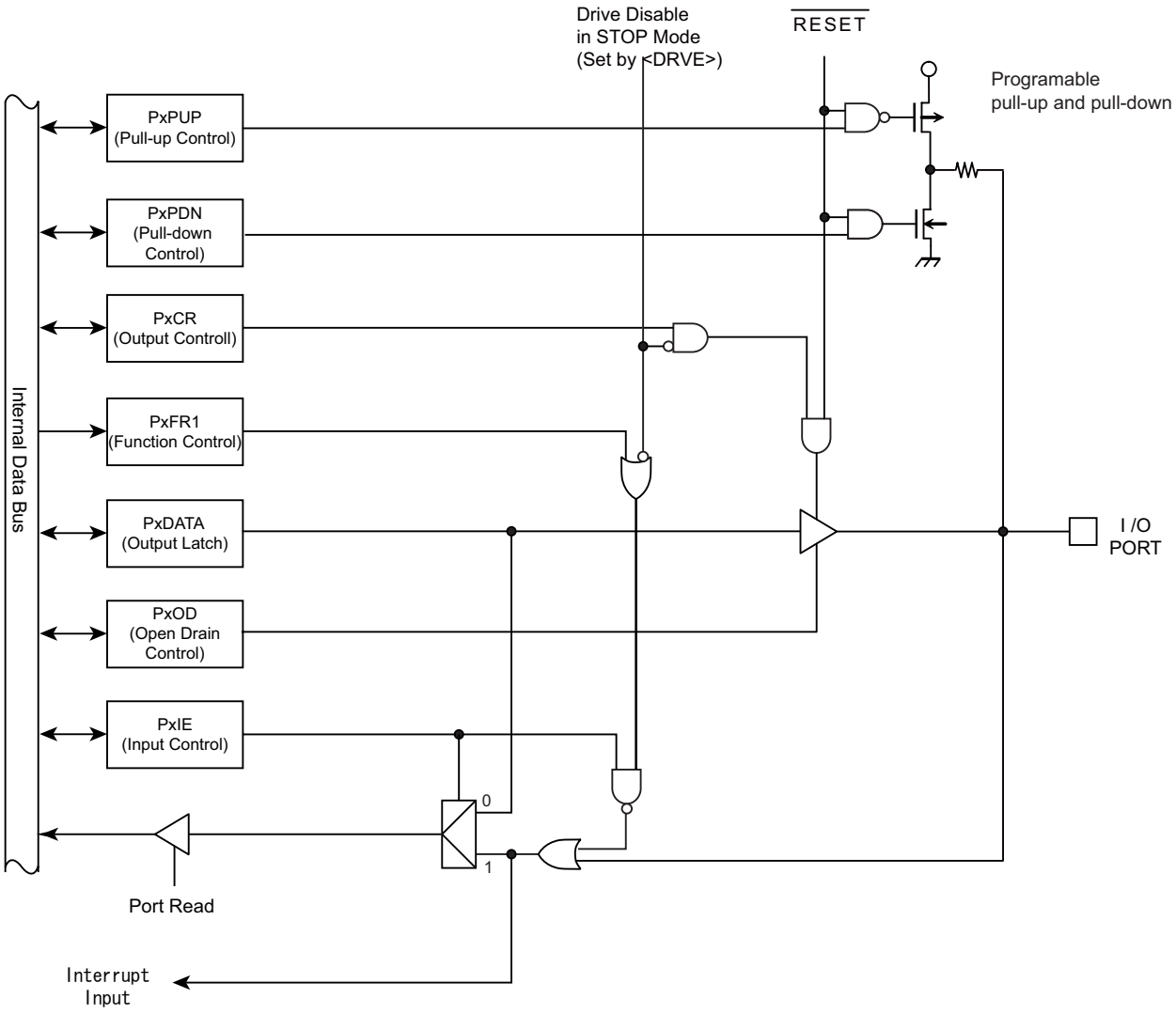


Figure 7-4 PORT Type T4

7.3.6 Type T5

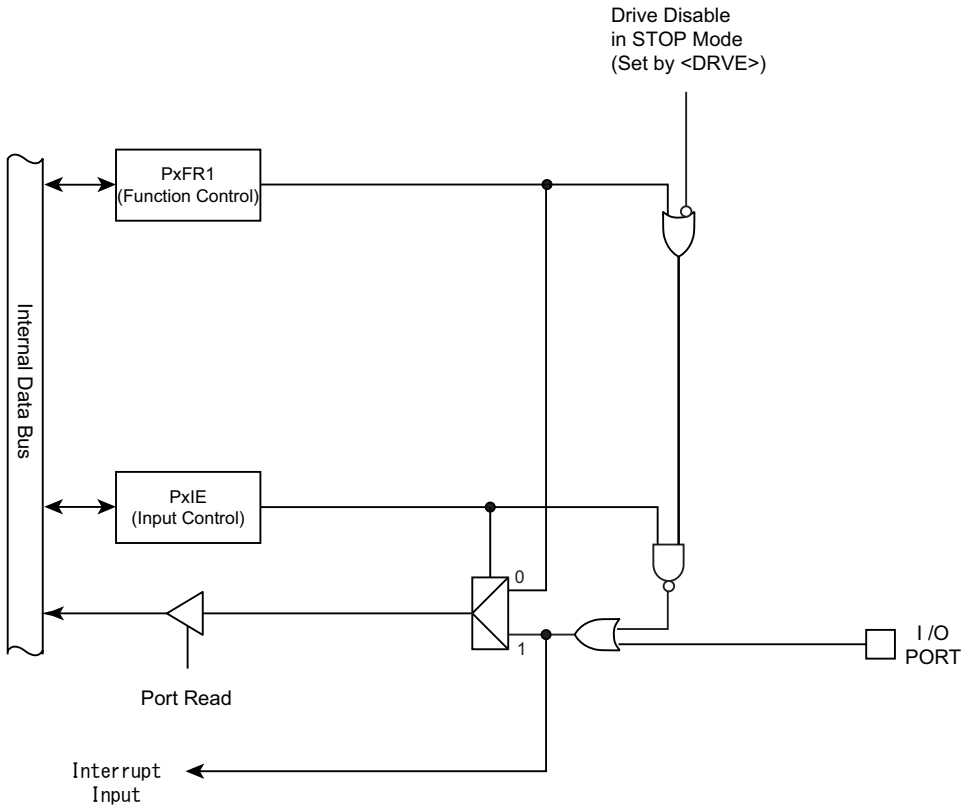


Figure 7-5 PORT Type T5

7.3.7 Type T6

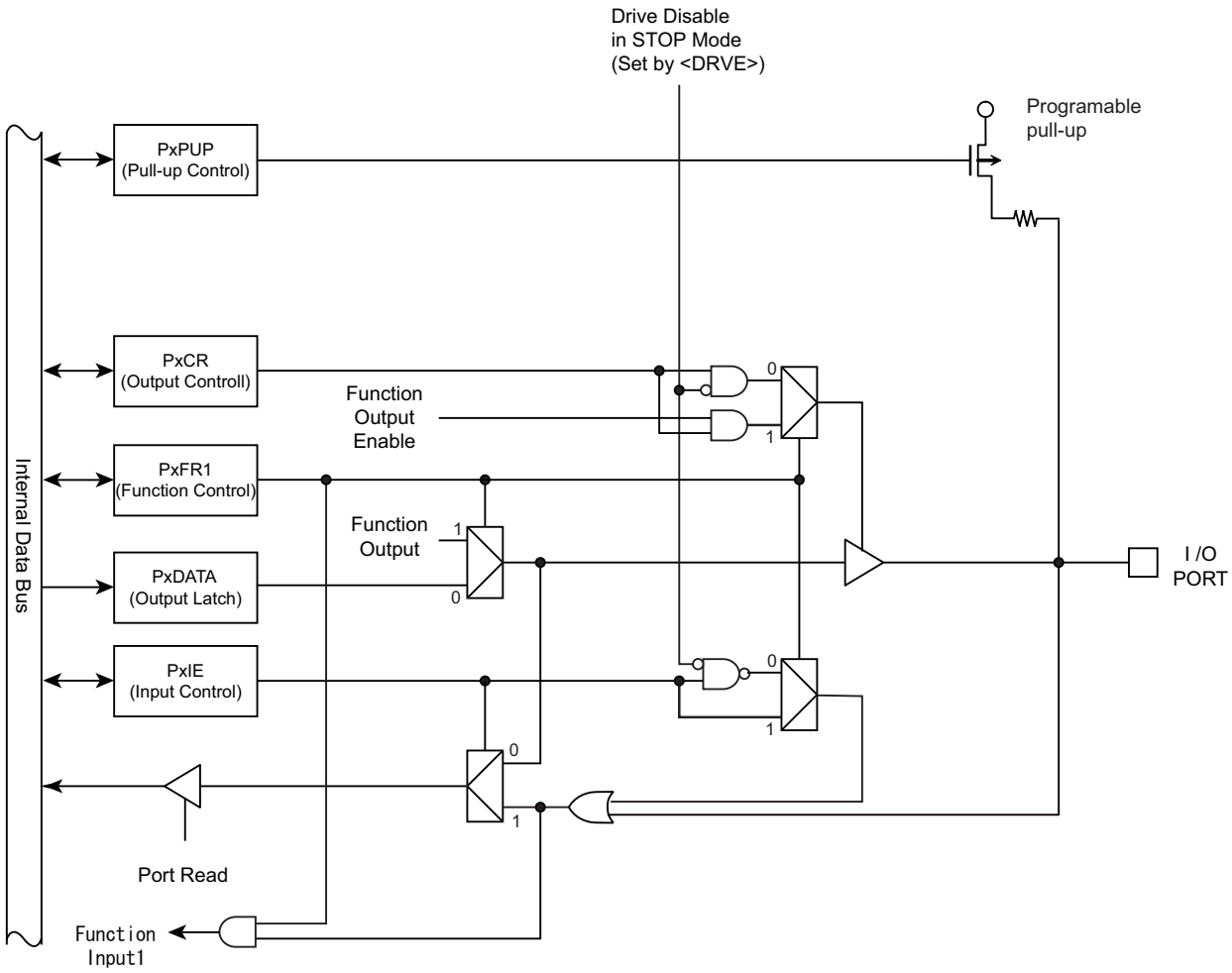


Figure 7-6 PORT Type T6

7.3.8 Type T7

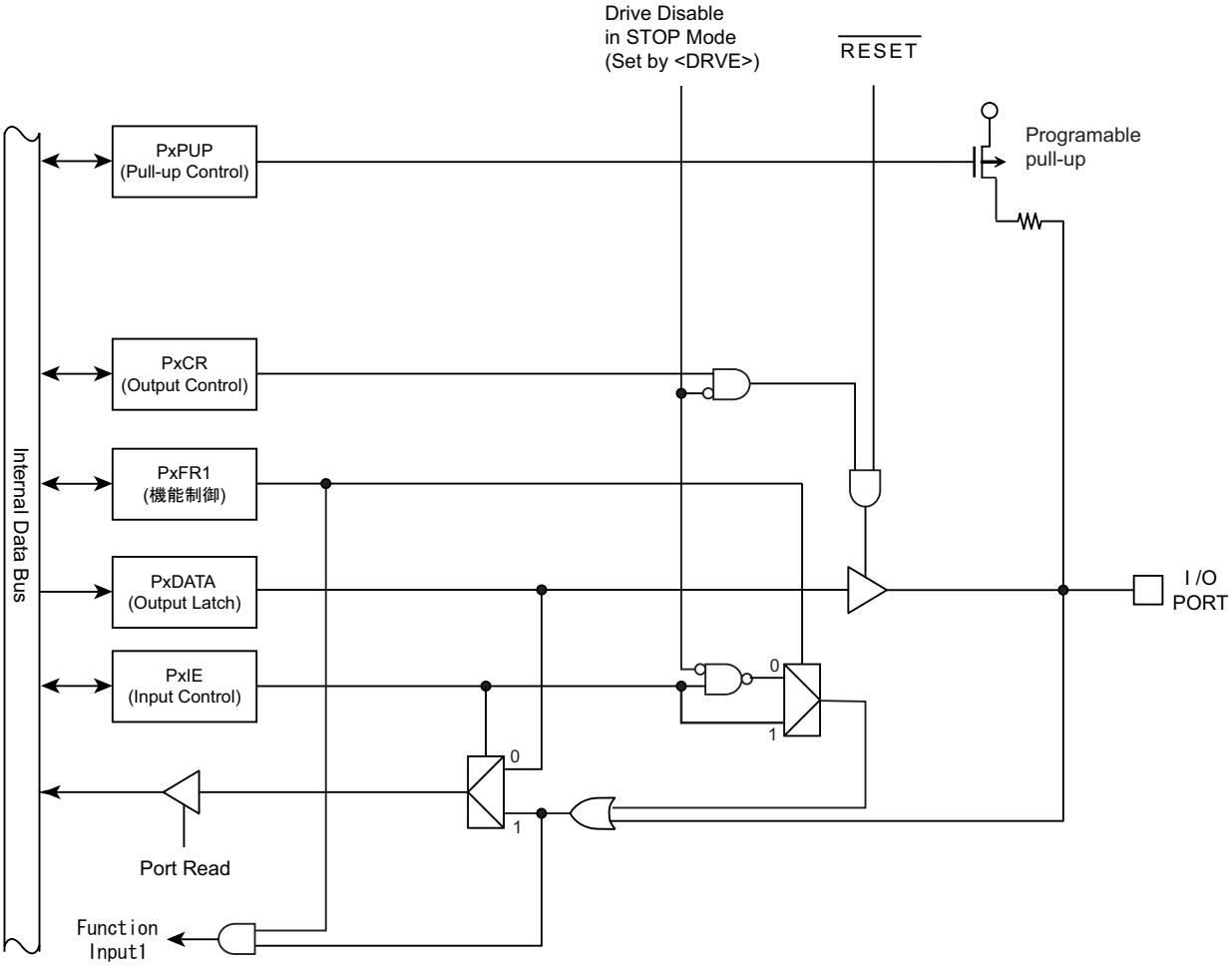


Figure 7-7 PORT Type T7

7.3.9 Type T8

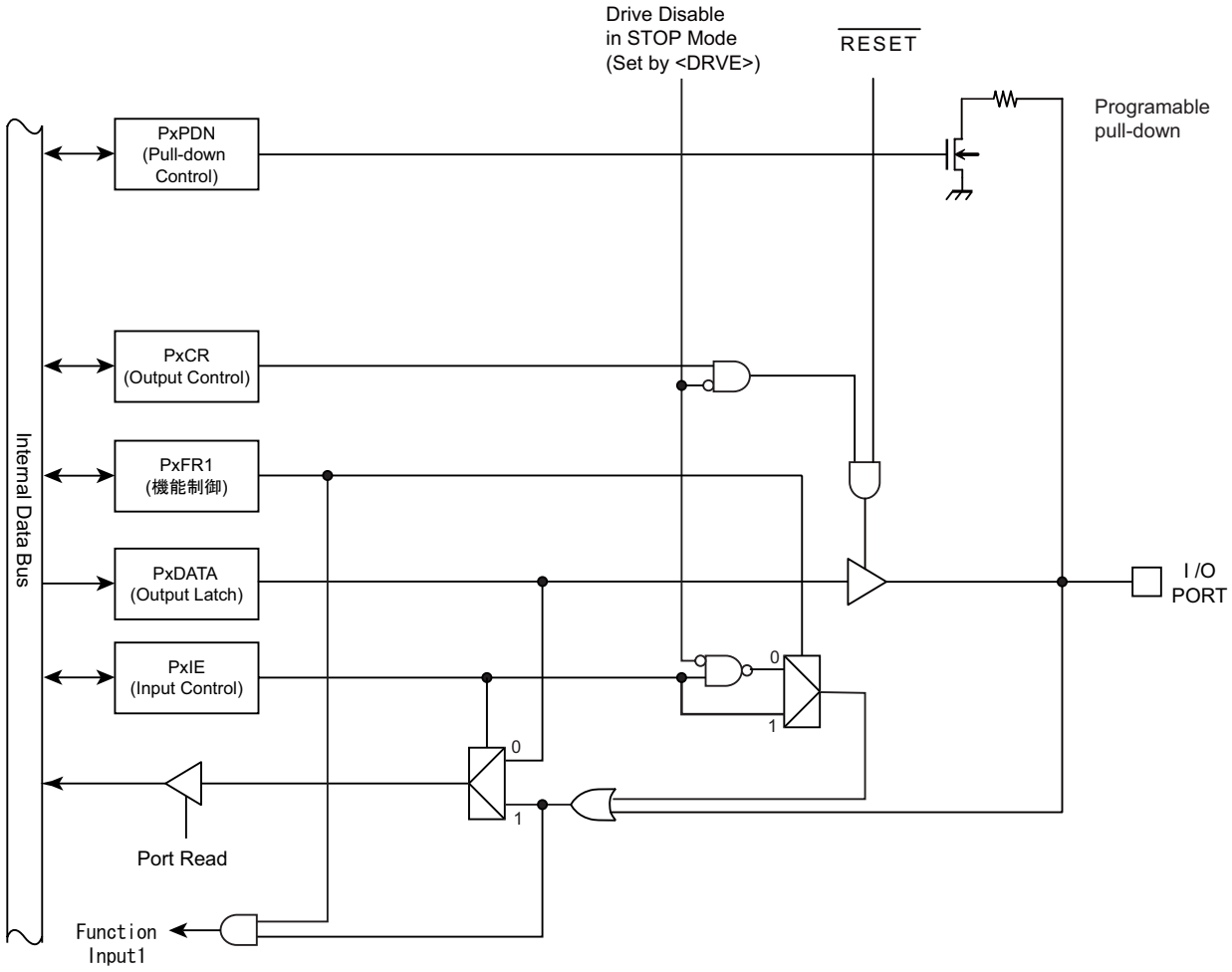


Figure 7-8 PORT Type T8

7.3.10 Type T9

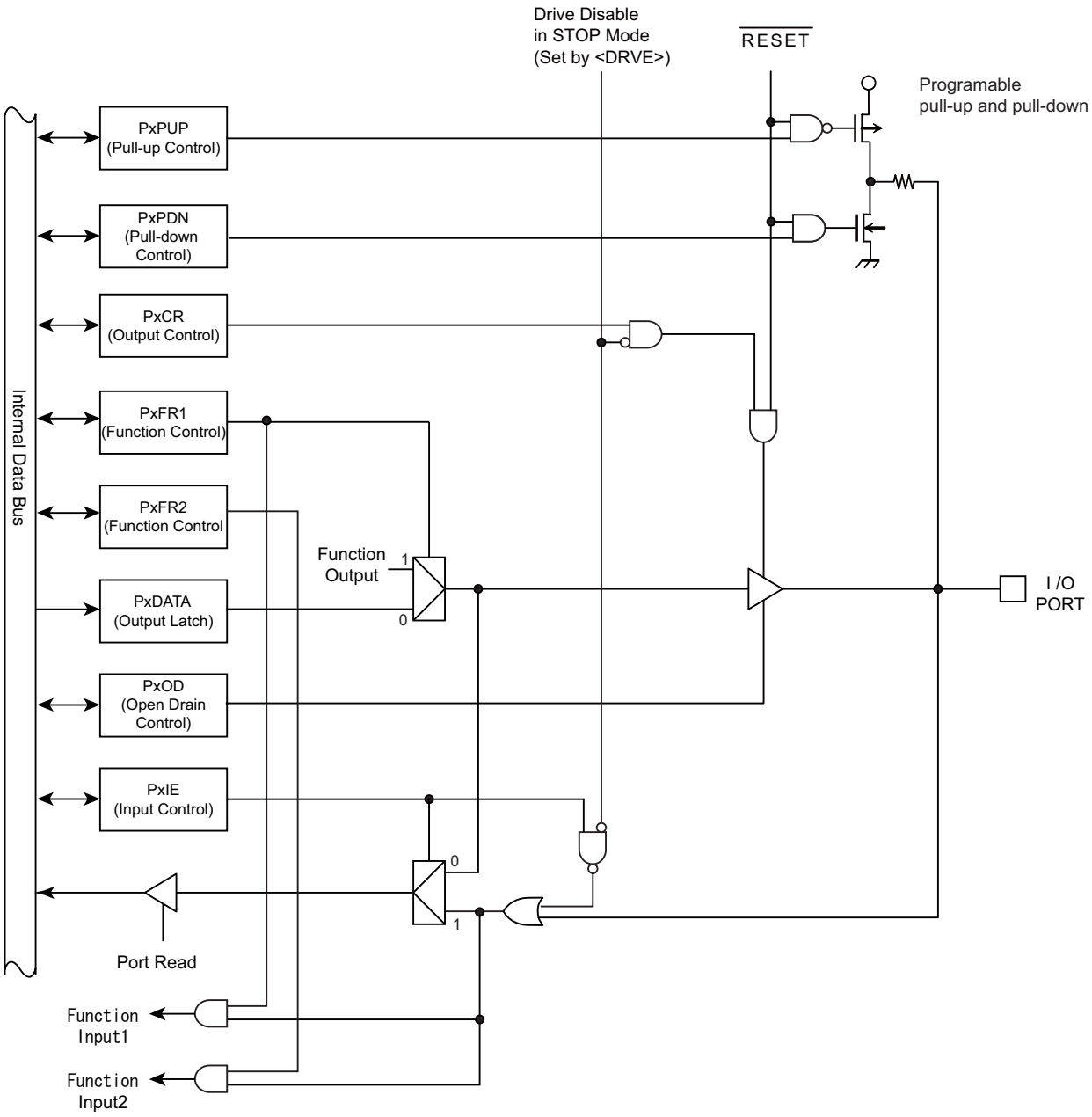


Figure 7-9 PORT Type T9

7.3.11 Type T10

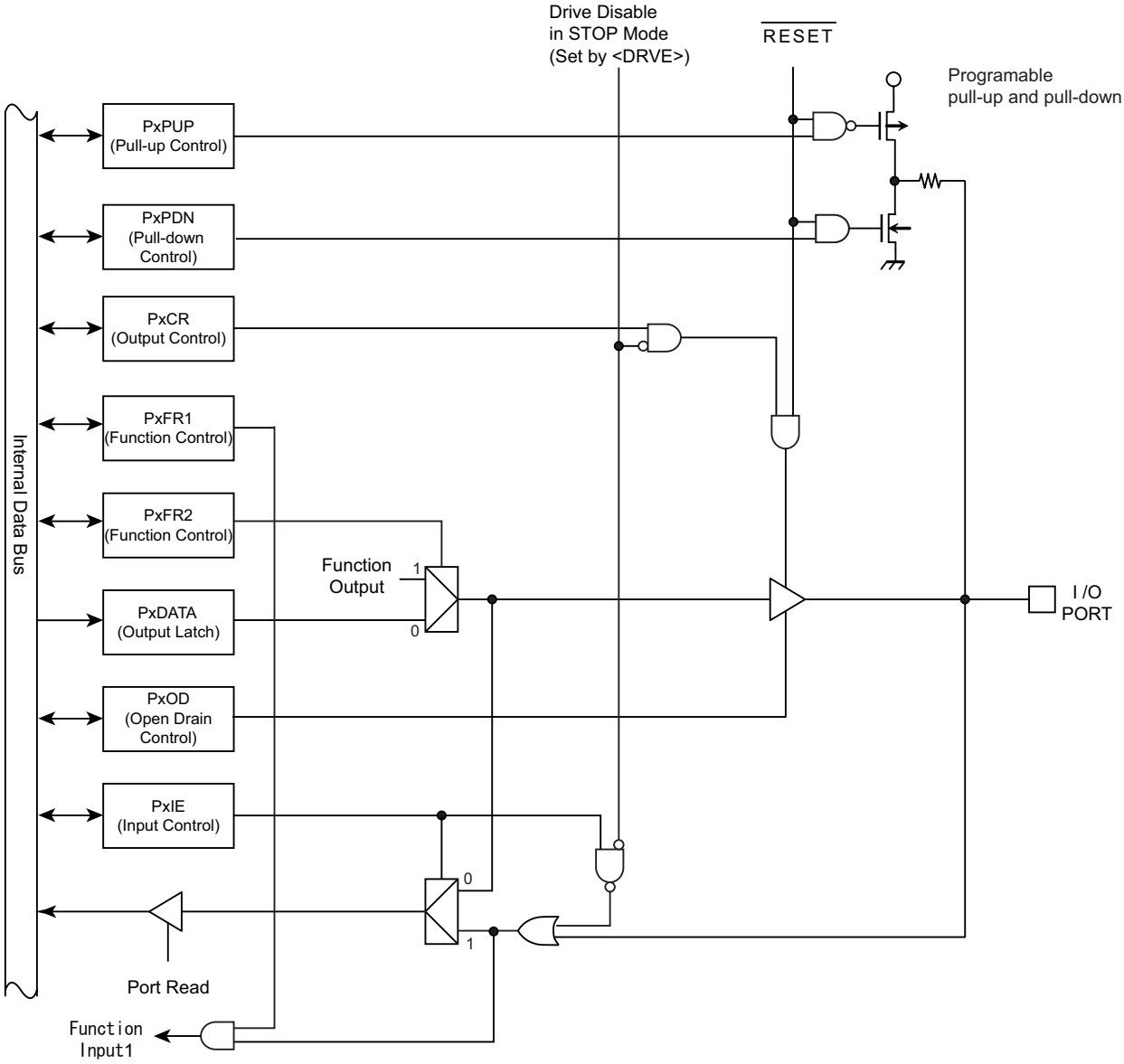


Figure 7-10 PORT Type T10

7.3.12 Type T11

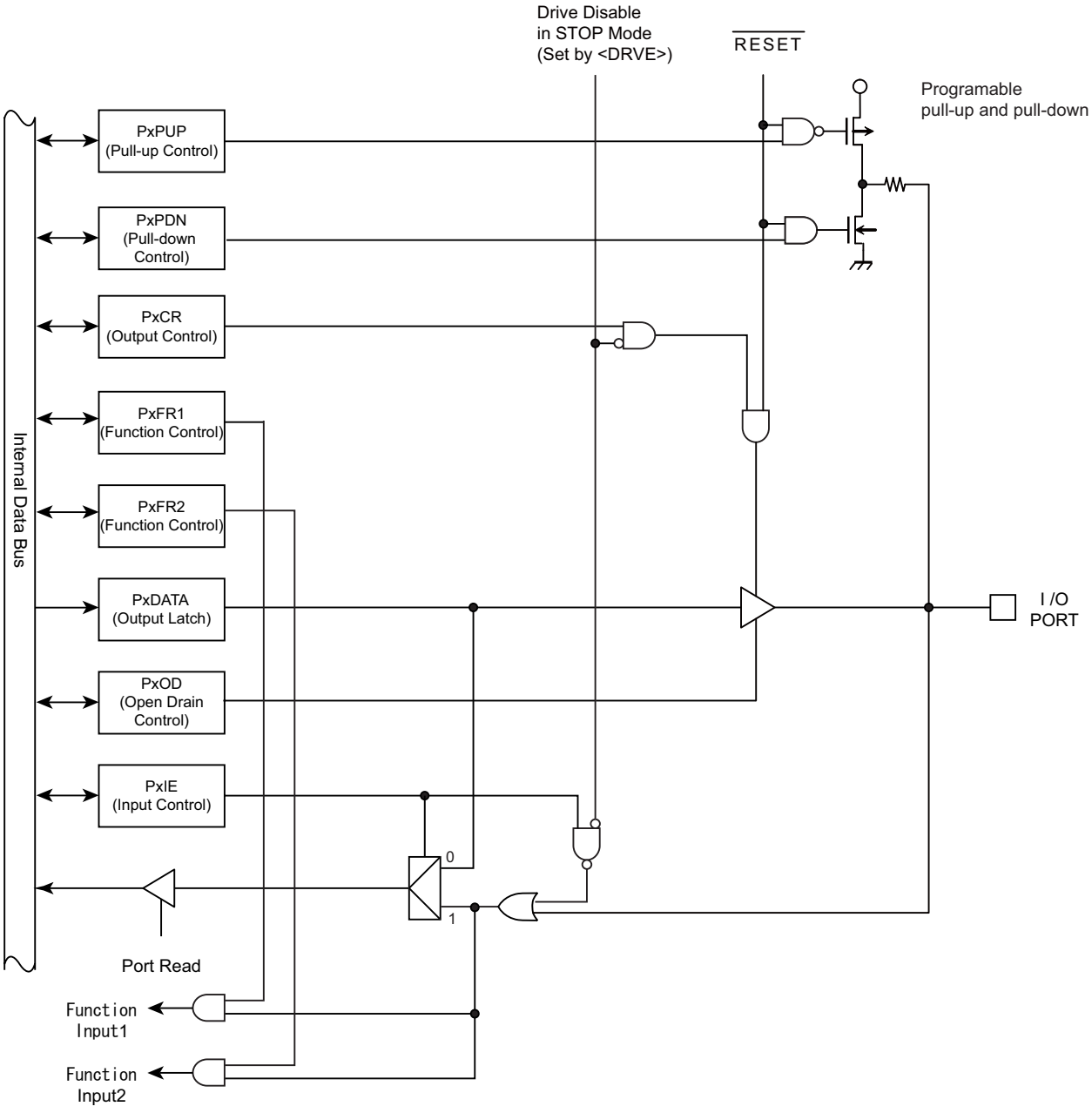


Figure 7-11 PORT Type T11

7.3.13 Type T12

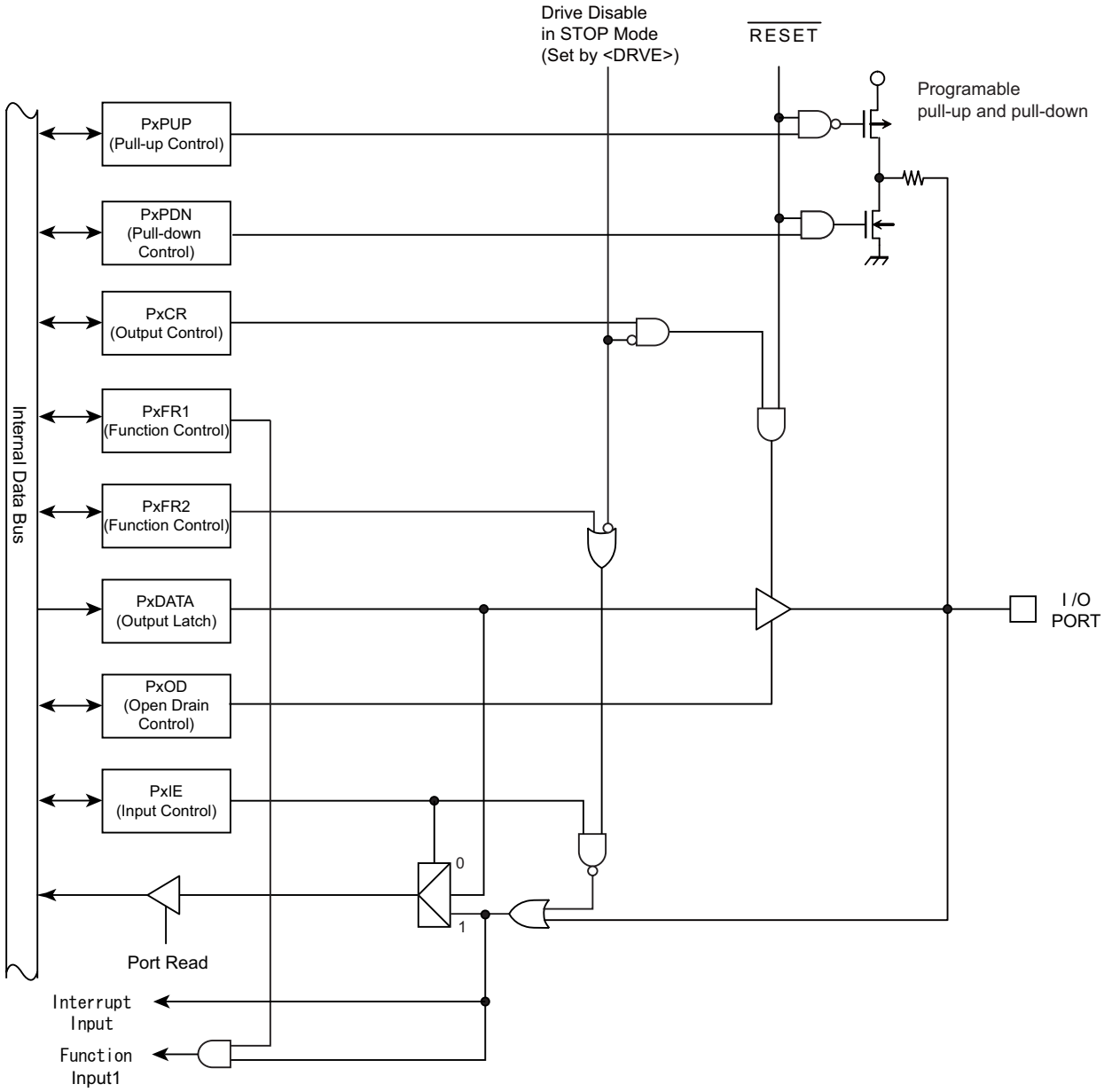


Figure 7-12 PORT Type T12

7.3.14 Type T13

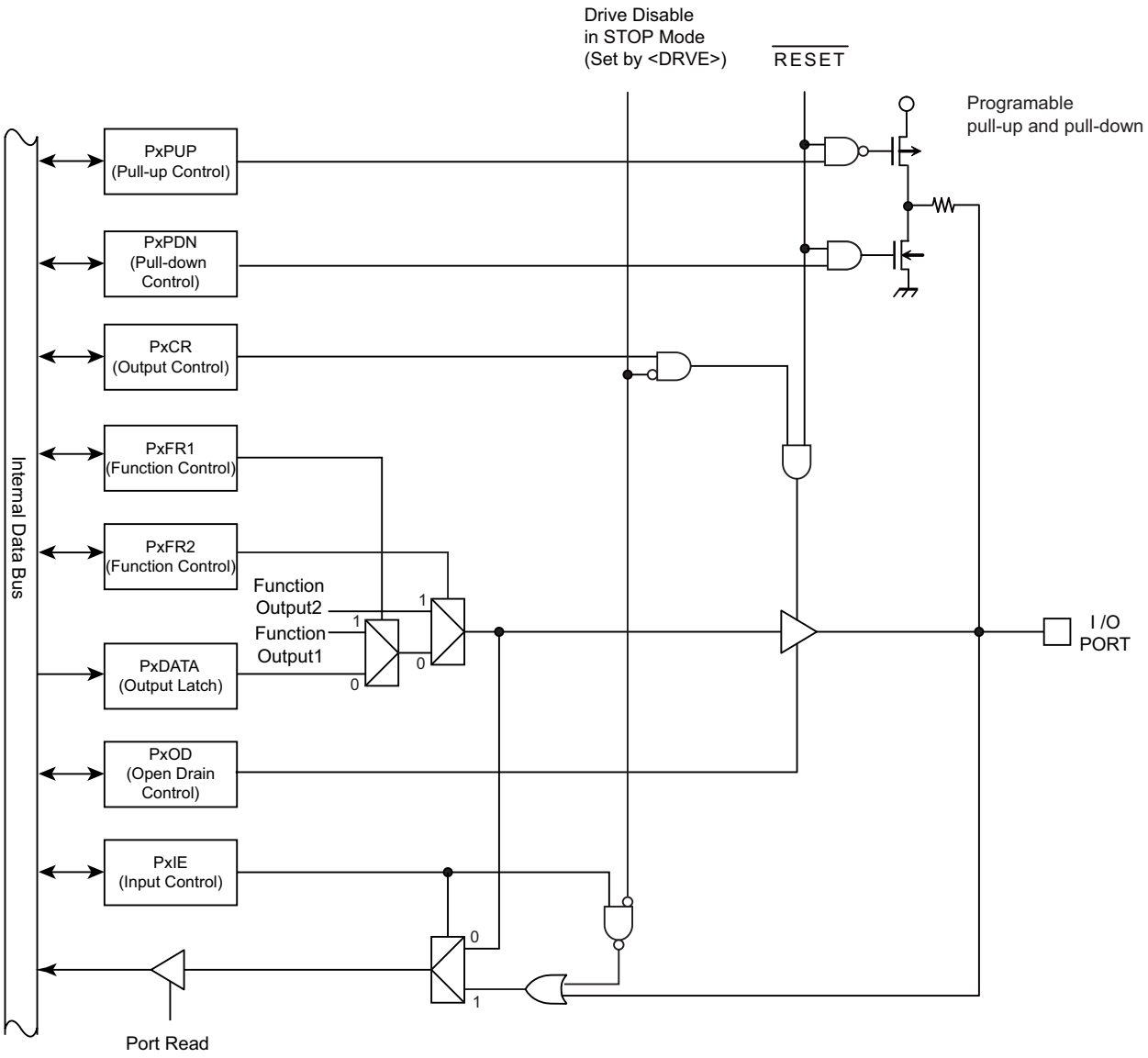


Figure 7-13 PORT Type T13

7.3.15 Type T14

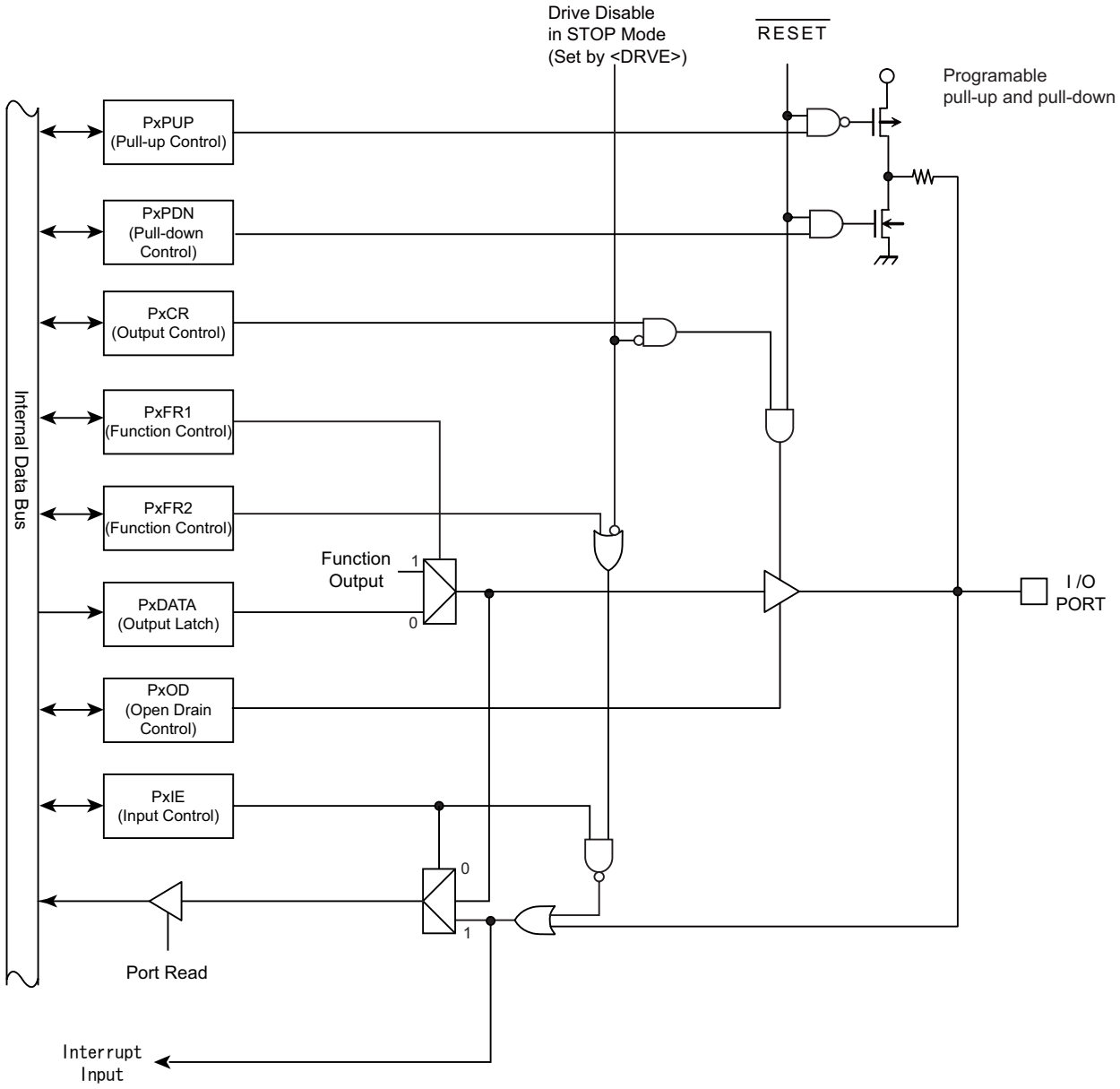


Figure 7-14 PORT Type T14

7.3.16 Type T15

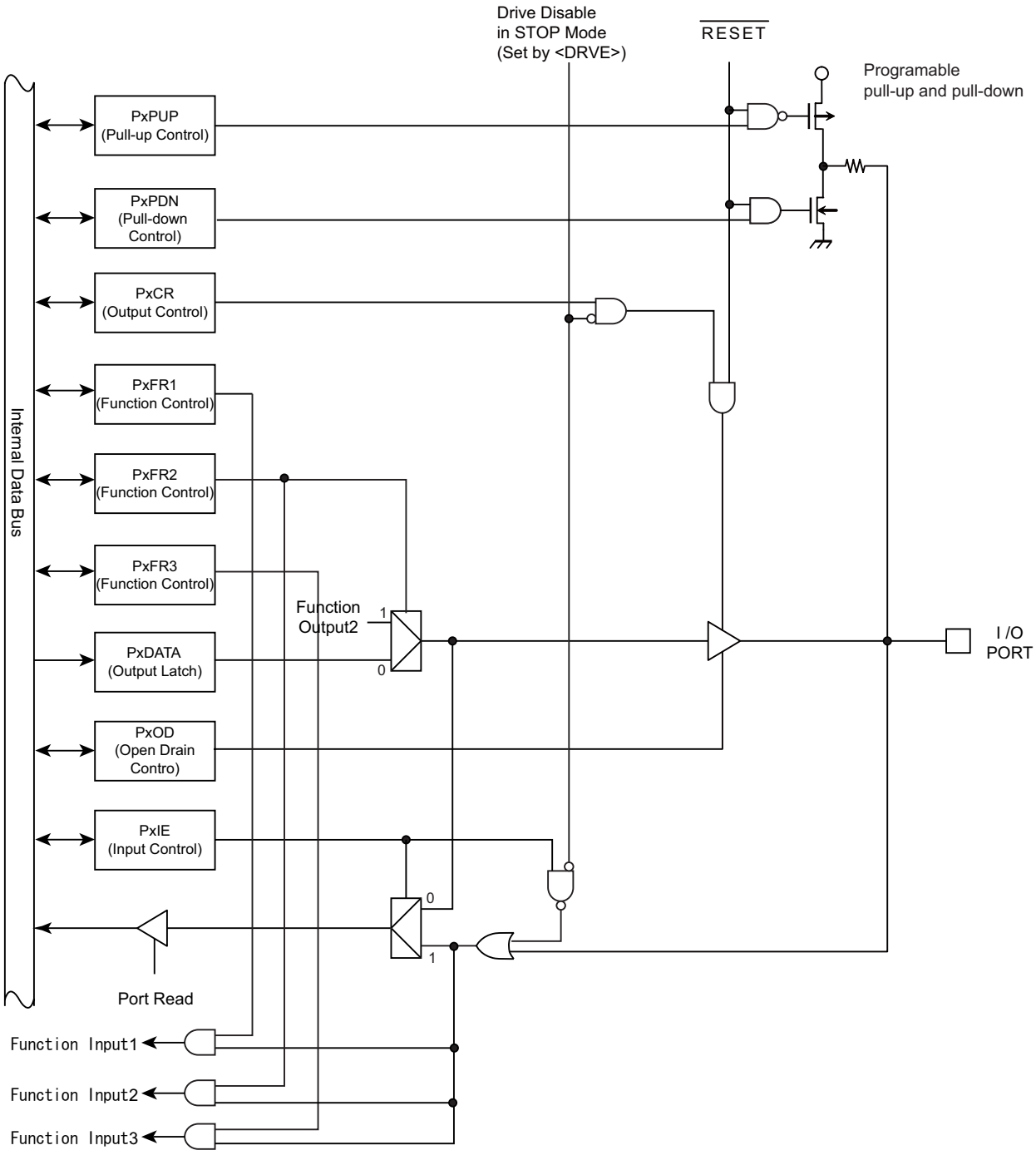


Figure 7-15 PORT Type T15

7.3.17 Type T16

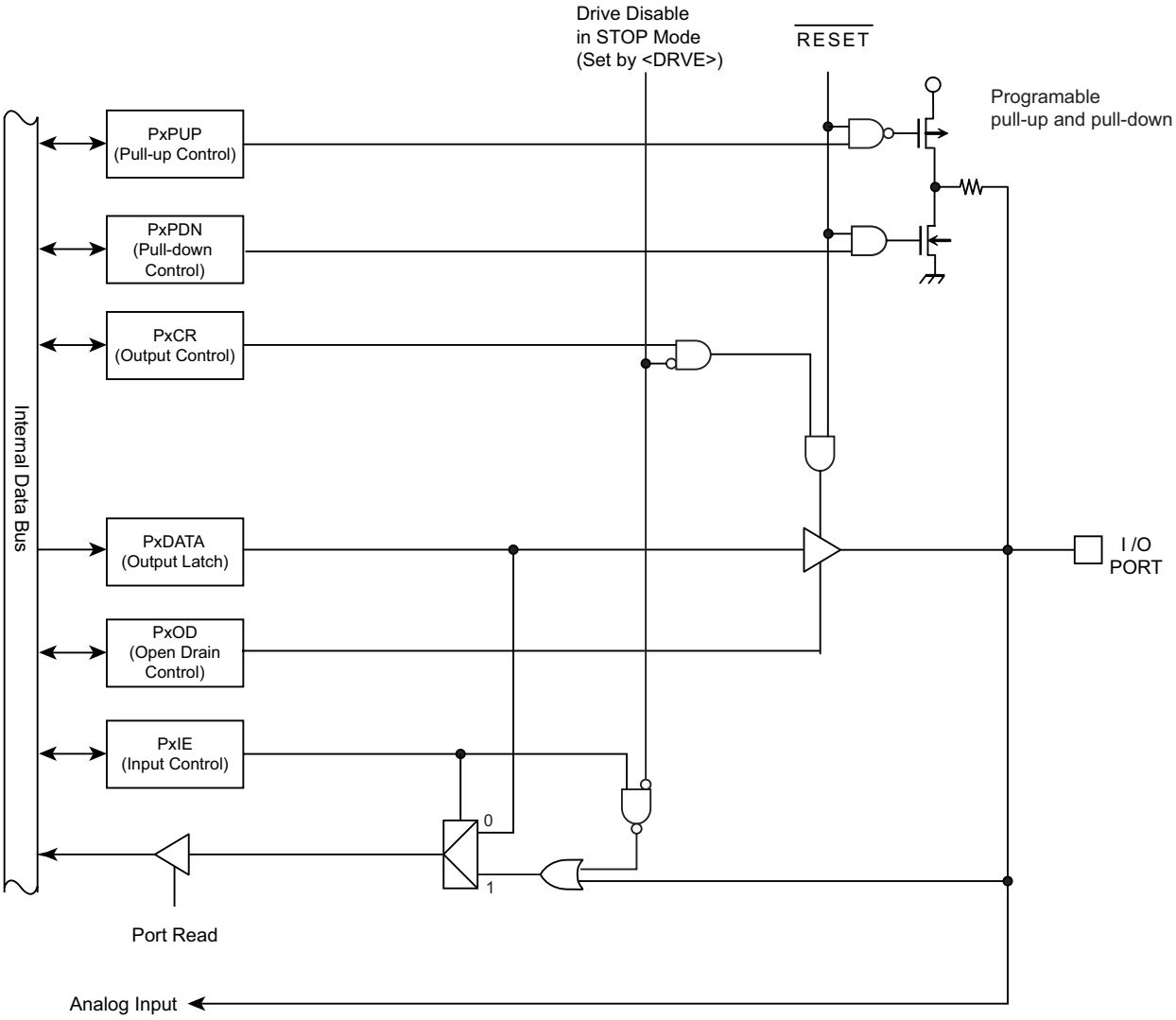


Figure 7-16 PORT Type T16

7.3.18 Type T17

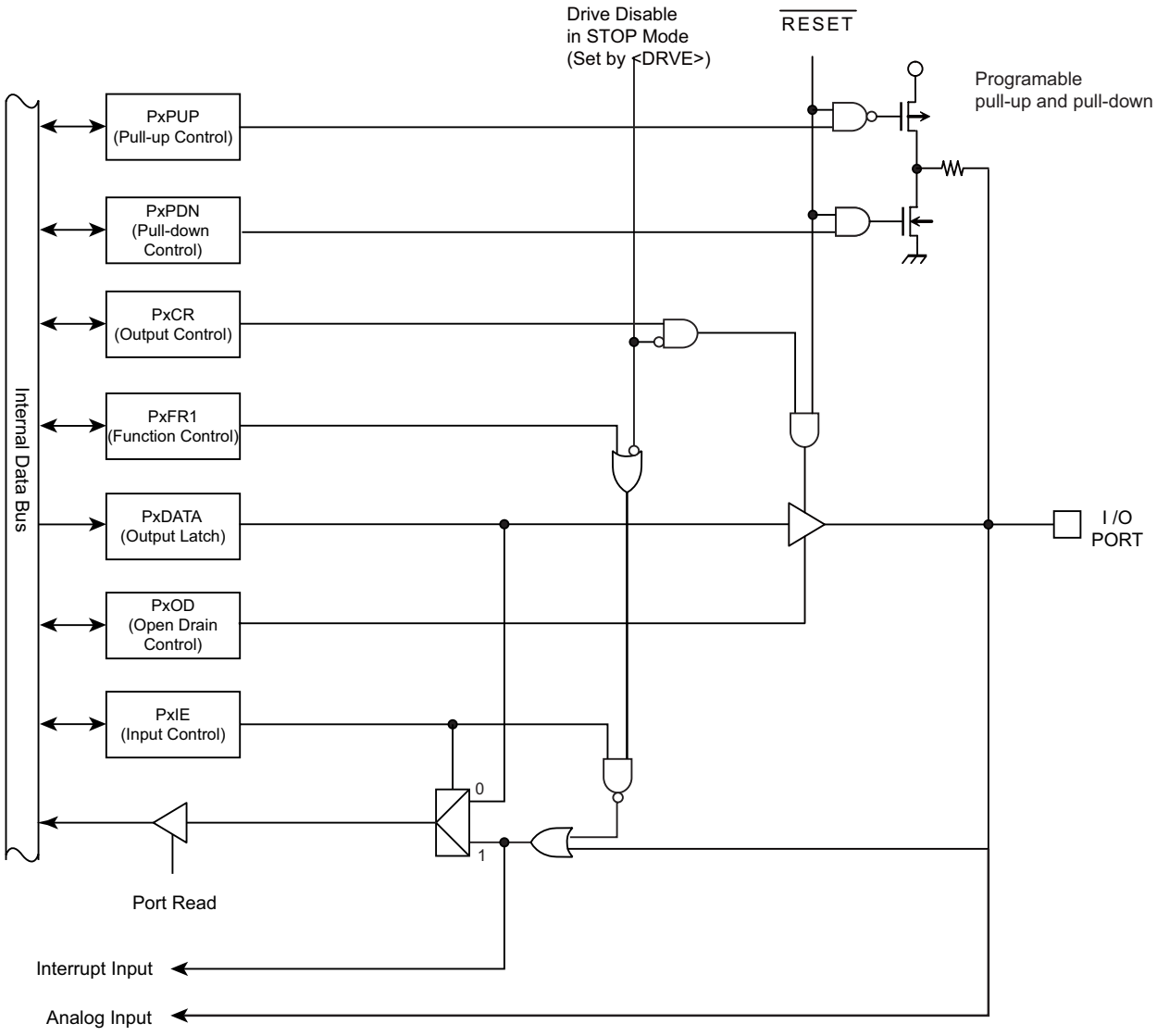


Figure 7-17 PORT Type T17

7.3.19 Type T18

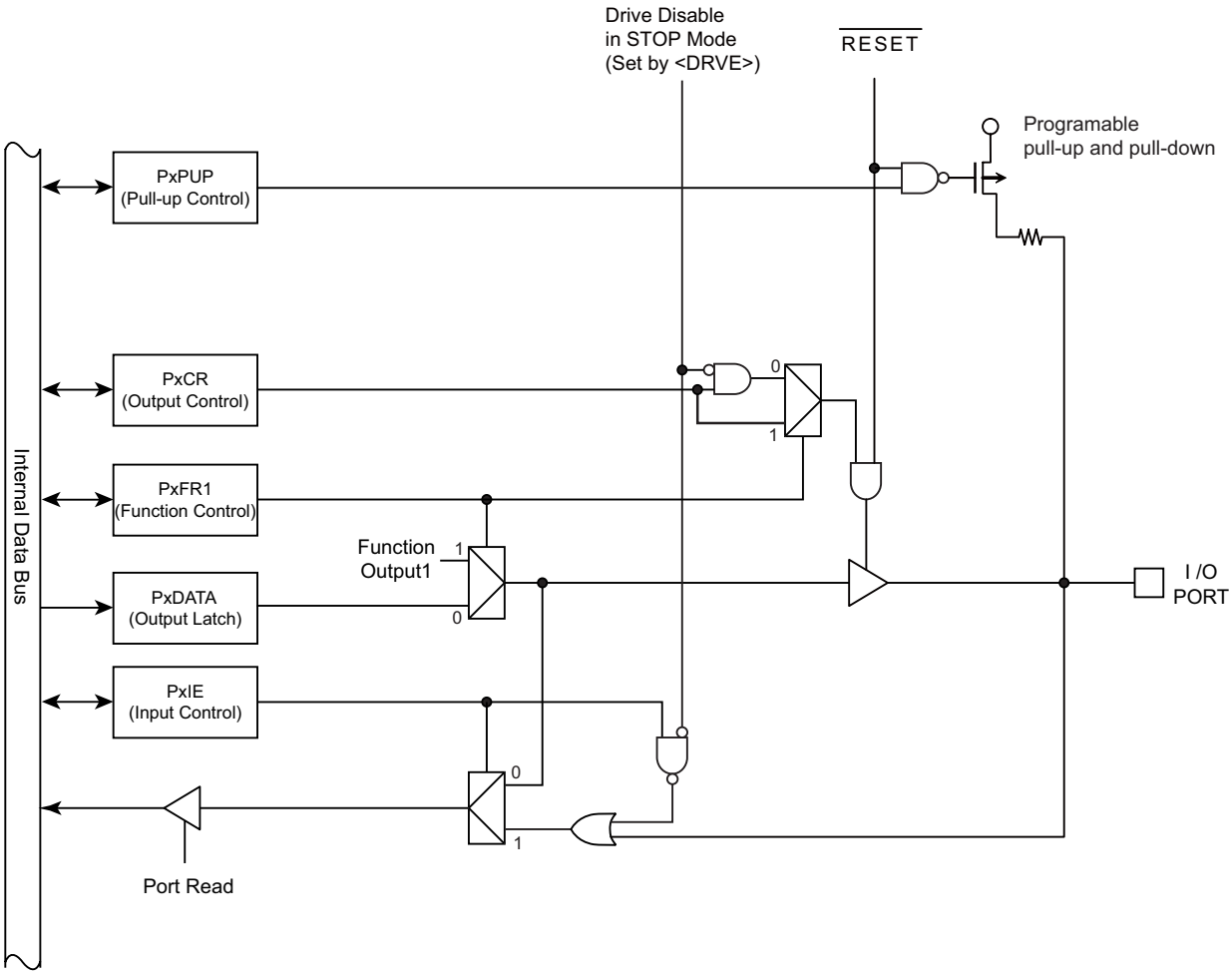


Figure 7-18 PORT Type T18

7.3.20 Type T19

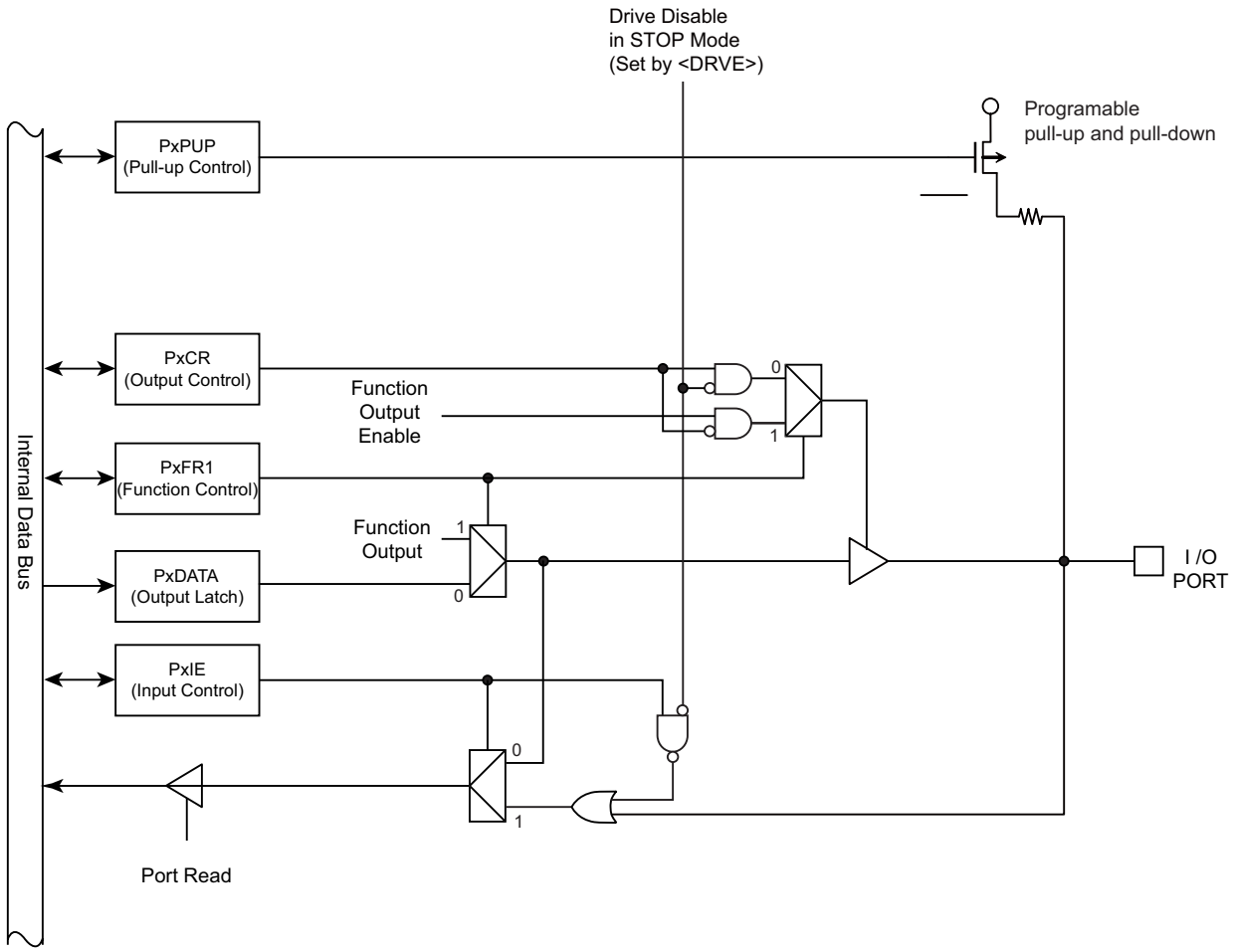
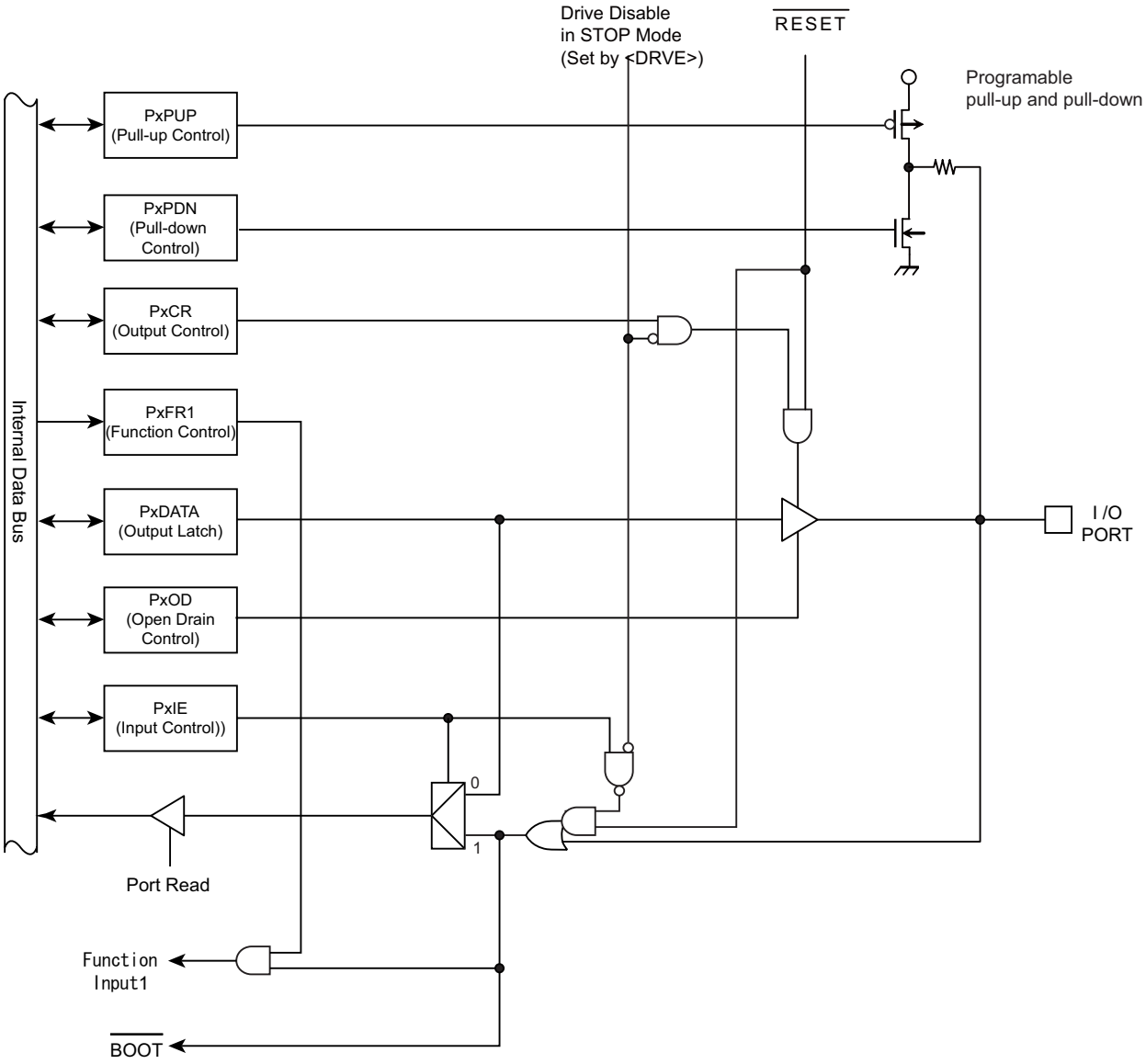
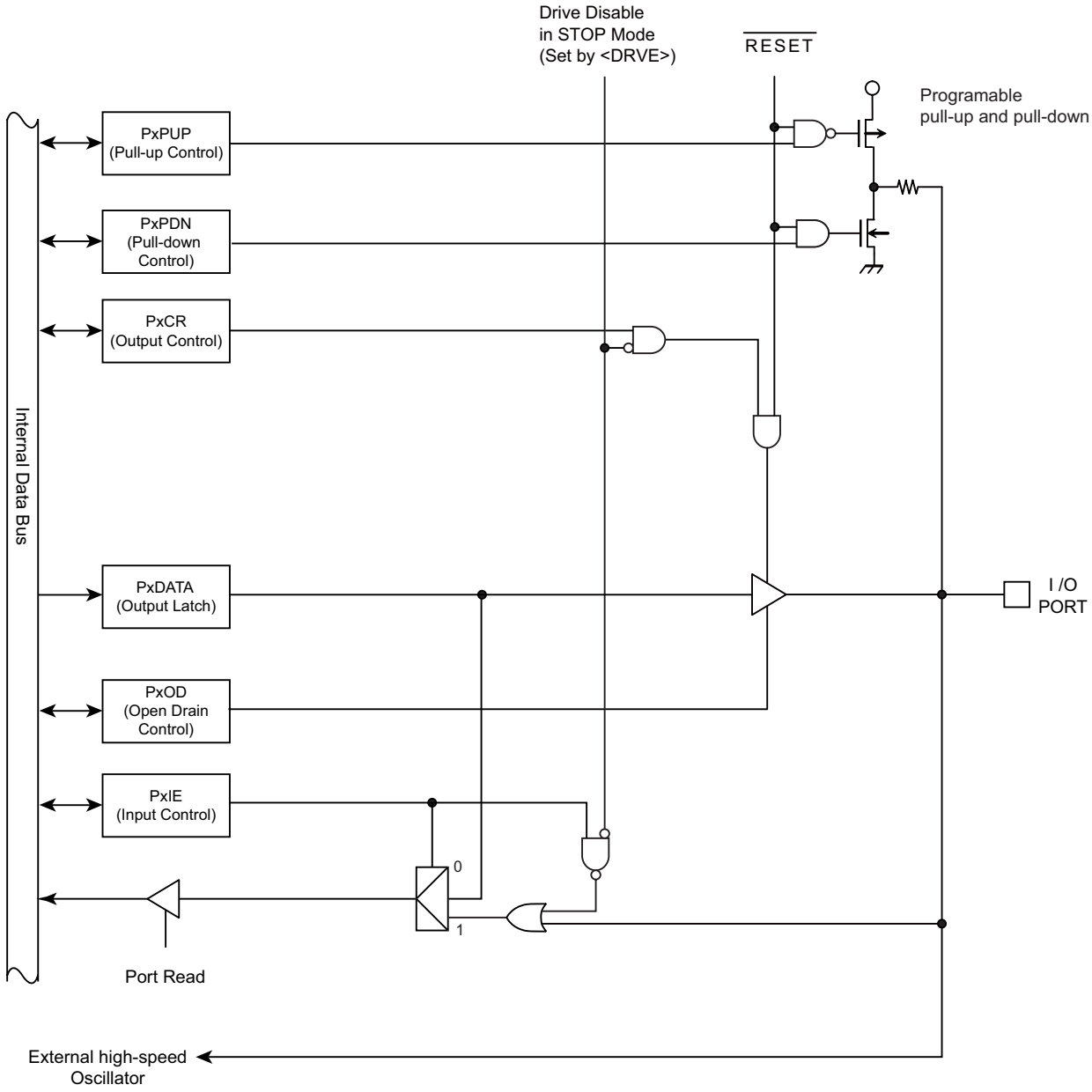


Figure 7-19 PORT Type T19

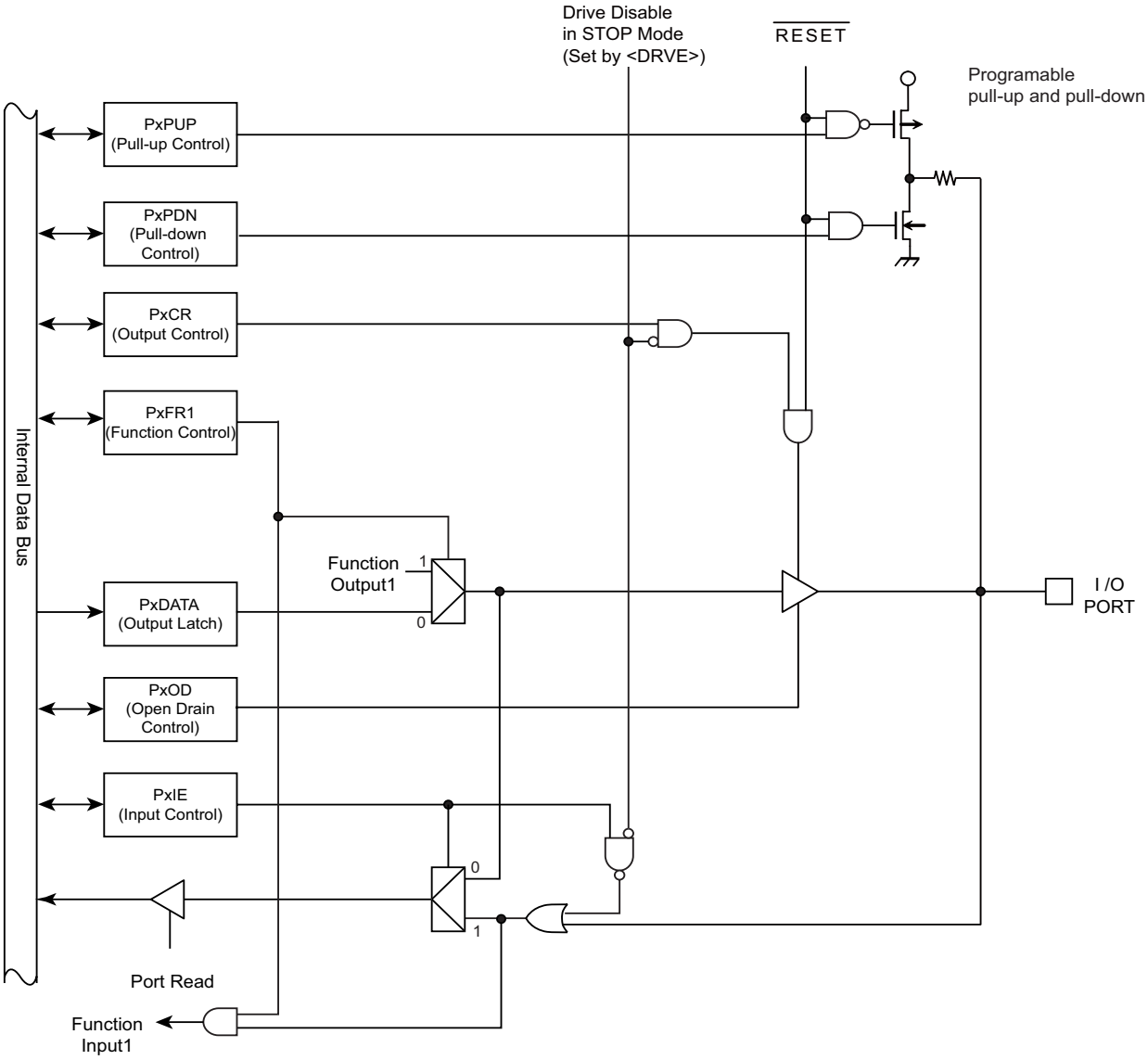
7.3.21 Type T20



7.3.22 Type T21



7.3.23 Type T22



7.4 Appendix Port Setting List

The following table shows the register setting for each function.

Initialization of the ports where the [●]does not exist in the "After reset" field is set to "0" for all register settings. Setting for the bit "x" can be arbitrarily-specified.

7.4.1 Port A Setting

Table 7-4 Port Setting List(Port A)

Pin	Port Type	Function	After reset	PACR	PAFR1	PAFR2	PAOD	PAPUP	PAPDN	PAIE
PA0	T12	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TB0IN (input)		0	1	0	x	x	x	1
		INT7 (Input)		0	0	1	x	x	x	1
PA1	T2	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		TB0OUT(Output)		1	1	-	x	x	x	0
PA2	T12	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TB1IN (Input)		0	1	0	x	x	x	1
		INT4 (Input)		0	0	1	x	x	x	1
PA3	T2	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		TB1OUT(output)		1	1	-	x	x	x	0
PA4	T9	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		SCLK1 (I / O)		1	1	0	x	x	x	1
		$\overline{\text{CTS1}}$ (Input)		0	0	1	x	x	x	1
PA5	T13	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TXD1 (Output)		1	1	0	x	x	x	0
		TB6OUT(Output)		1	0	1	x	x	x	0
PA6	T11	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		RXD1 (Input)		0	1	0	x	x	x	1
		TB6IN (Input)		0	0	1	x	x	x	1
PA7	T12	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TB4IN (Input)		0	1	0	x	x	x	1
		INT8 (Input)		0	0	1	x	x	x	1

7.4.2 Port B Setting

Table 7-5 Port Setting List(Port B)

Pin	Port Type	Function	After reset	PBCR	PBFR1	PBOD	PBPUP	PBPDN	PBIE
PB0	T18	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TRACECLK (Output)		1	1	0	0	0	0
PB1	T18	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TRACEDATA0 (Output)		1	1	0	0	0	0
PB2	T18	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TRACEDATA1 (Output)		1	1	0	0	0	0
PB3	T6	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TMS / SWDIO (I / O)		1	1	0	1	0	1
PB4	T8	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TCK / SWCLK (Input)		0	1	0	0	1	1
PB5	T19	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TDO / SWV (Output)		1	1	0	0	0	0
PB6	T7	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TDI (Input)		0	1	0	1	0	1
PB7	T7	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		$\overline{\text{TRST}}$ (Input)		0	1	0	1	1	1

7.4.3 Port C Setting

Table 7-6 Port Setting List(Port C)

Pin	Port Type	Function	After reset	PCCR	PCFR1	PCOD	PCPUP	PCPDN	PCIE
PC0	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		UO0 (Output)		1	1	x	x	x	0
PC1	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		XO0 (Output)		1	1	x	x	x	0
PC2	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		VO0 (Output)		1	1	x	x	x	0
PC3	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		YO0 (Output)		1	1	x	x	x	0
PC4	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		WO0 (Output)		1	1	x	x	x	0
PC5	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		ZO0 (Output)		1	1	x	x	x	0
PC6	T3	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		$\overline{\text{EMG0}}$ (Input)		0	1	x	x	x	1
PC7	T3	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		$\overline{\text{OVV0}}$ (Input)		0	1	x	x	x	1

7.4.4 Port D Setting

Table 7-7 Port Setting List(Port D)

Pin	Port Type	Function	After reset	PDCR	PDFR1	PDFR2	PDOD	PDPUP	PDPDN	PDIE
PD0	T11	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		ENCA0 (Input)		0	1	0	x	x	x	1
		TB5IN (Input)		0	0	1	x	x	x	1
PD1	T10	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		ENCB0 (Input)		0	1	0	x	x	x	1
		TB5OUT (Output)		1	0	1	x	x	x	0
PD2	T3	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		ENCZ0(Input)		0	1	-	x	x	x	1
PD3	T4	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		INT9 (Input)		0	1	-	x	x	x	1
PD4	T9	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		SCLK2 (I / O)		1	1	0	x	x	x	1
		$\overline{\text{CTS2}}$ (Input)		0	0	1	x	x	x	1
PD5	T2	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		TXD2 (Output)		1	1	-	x	x	x	0
PD6	T3	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		RXD2 (Input)		0	1	-	x	x	x	1

7.4.5 Port E Setting

Table 7-8 Port Setting List(Port E)

Pin	Port Type	Function	After reset	PECR	PEFR1	PEFR2	PEOD	PEPUP	PEPDN	PEIE
PE0	T2	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		TXD0 (Output)		1	1	-	x	x	x	0
PE1	T3	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		RXD0 (Input)		0	1	-	x	x	x	1
PE2	T9	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		SCLK0 (I / O)		1	1	0	x	x	x	1
		$\overline{\text{CTS0}}$ (Input)		0	0	1	x	x	x	1
PE3	T2	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		TB4OUT (Output)		1	1	-	x	x	x	0
PE4	T12	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TB2IN (Input)		0	1	0	x	x	x	1
		INT5 (Input)		0	0	1	x	x	x	1
PE5	T2	Input Port		0	0	-	x	x	x	1
		Output Port		1	0	-	x	x	x	0
		TB2OUT (Output)		1	1	-	x	x	x	0
PE6	T12	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TB3IN (Input)		0	1	0	x	x	x	1
		INT6 (Input)		0	0	1	x	x	x	1
PE7	T14	Input Port		0	0	0	x	x	x	1
		Output Port		1	0	0	x	x	x	0
		TB3OUT (Output)		1	1	0	x	x	x	0
		INT7 (Input)		0	0	1	x	x	x	1

7.4.6 Port F Setting

Table 7-9 Port Setting List(Port F)

Pin	Port Type	Function	After reset	PFCR	PFFR1	PFFR2	PFFR3	PFOD	PFPUP	PFPDN	PFIE
PF0	T20	Input Port		0	0	-	-	x	x	x	1
		Output Port		1	0	-	-	x	x	x	0
		TB7IN (Input)		0	1	-	-	x	x	x	1
PF1	T2	Input Port		0	0	-	-	x	x	x	1
		Output Port		1	0	-	-	x	x	x	0
		TB7OUT (Output)		1	1	-	-	x	x	x	0
PF2	T15	Input Port		0	0	0	0	x	x	x	1
		Output Port		1	0	0	0	x	x	x	0
		ENCA1 (Input)		0	1	0	0	x	x	x	1
		SCLK3 (I / O)		1	0	1	0	x	x	x	1
		$\overline{\text{CTS3}}$ (Input)		0	0	0	1	x	x	x	1
PF3	T10	Input Port		0	0	0	-	x	x	x	1
		Output Port		1	0	0	-	x	x	x	0
		ENCB1 (Input)		0	1	0	-	x	x	x	1
		TXD3 (Output)		1	0	1	-	x	x	x	0
PF4	T11	Input Port		0	0	0	-	x	x	x	1
		Output Port		1	0	0	-	x	x	x	0
		ENCZ1 (Input)		0	1	0	-	x	x	x	1
		RXD3 (Input)		0	0	1	-	x	x	x	1

Note: The PF0 input and pull-up are enabled and act as $\overline{\text{BOOT}}$ input pin while a $\overline{\text{RESET}}$ is in "Low" state

7.4.7 Port G Setting

Table 7-10 Port Setting List(Port G)

Pin	Port Type	Function	After reset	PGCR	PGFR1	PGOD	PGPUP	PGPDN	PGIE
PG0	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		UO1 (Output)		1	1	x	x	x	0
PG1	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		XO1 (Output)		1	1	x	x	x	0
PG2	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		VO1 (Output)		1	1	x	x	x	0
PG3	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		YO1 (Output)		1	1	x	x	x	0
PG4	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		WO1 (Output)		1	1	x	x	x	0
PG5	T1	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		ZO1 (Output)		1	1	x	x	x	0
PG6	T3	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		$\overline{\text{EMG1}}$ (Input)		0	1	x	x	x	1
PG7	T3	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		$\overline{\text{OVV1}}$ (Input)		0	1	x	x	x	1

7.4.8 Port H Setting

Table 7-11 Port Setting List(Port H)

Pin	Port Type	Function	After reset	PHCR	PHFR1	PHOD	PHPUP	PHPDN	PHIE
PH0	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INT0 (input)		0	1	x	x	x	1
PH1	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INT1 (Input)		0	1	x	x	x	1
PH2	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INT2 (Input)		0	1	x	x	x	1
PH3	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PH4	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PH5	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PH6	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PH7	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0

7.4.9 Port I Setting

Table 7-12 Port Setting List(Port I)

Pin	Port Type	Function	After reset	PICR	PIOD	PIPUP	PIPDN	PIIE
PI0	T16	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
		Analog Input		0	0	0	0	0
PI1	T16	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
		Analog Input		0	0	0	0	0
PI2	T16	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
		Analog Input		0	0	0	0	0
PI3	T16	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
		Analog Input		0	0	0	0	0

7.4.10 Port J Setting

Table 7-13 Port Setting List(Port J)

Pin	Port Type	Function	After reset	PJCR	PJFR1	PJOD	PJPUP	PJPDN	PJIE
PJ0	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PJ1	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PJ2	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PJ3	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PJ4	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PJ5	T16	Input Port		0	-	x	x	x	1
		Output Port		1	-	x	x	x	0
		Analog Input		0	-	0	0	0	0
PJ6	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INTC (Input)		0	1	x	x	x	1
PJ7	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INTD (Input)		0	1	x	x	x	1

7.4.11 Port K Setting

Table 7-14 Port Setting List(Port K)

Pin	Port Type	Function	After reset	PKCR	PKFR1	PKOD	PKPUP	PKPDN	PKIE
PK0	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INTE (Input)		0	1	x	x	x	1
PK1	T17	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		Analog Input		0	0	0	0	0	0
		INTF (Input)		0	1	x	x	x	1

7.4.12 Port L Setting

Table 7-15 Port Setting List(Port L)

Pin	Port Type	Function	After reset	PLFR1	PLIE
PL0	T5	Input Port		0	1
		Output Port		0	0
		INTB (Input)		1	1
PL1	T5	Input Port		0	1
		Output Port		0	0
		INTA (Input)		1	1

7.4.13 Port M Setting

Table 7-16 Port Setting List(Port M)

Pin	Port Type	Function	After reset	PMCR	PMOD	PMPUP	PMPDN	PMIE
PM0	T21	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0
PM1	T21	Input Port		0	x	x	x	1
		Output Port		1	x	x	x	0

Note: X1,X2 exist

7.4.14 Port N Setting

Table 7-17 Port Setting List(Port N)

Pin	Port Type	Function	After reset	PNCR	PNFR1	PNOD	PNPUP	PNPDN	PNIE
PN0	T22	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		SO (O)		1	1	x	x	x	0
		SDA (I / O)		1	1	1	x	x	1
PN1	T22	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		SI (I)		0	1	x	x	x	1
		SCL (I / O)		1	1	1	x	x	1
PN2	T22	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		SCK (I / O)		1	1	x	x	x	1
PN3	T3	Input Port		0	0	x	x	x	1
		Output Port		1	0	x	x	x	0
		TB7IN (Input)		0	1	x	x	x	1

Note: When you set 1 to <PN3F1> (selected TB7IN), please set 0 to <PF0F1>. However, the input of PF0 becomes effective when 1 is set to both <PN3F1> and <PF0F1>.

8. 16-bit Timer / Event Counters (TMRB)

8.1 Outline

TMRB operate in the following four operation modes:

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

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

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

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

8.2 Differences in the Specifications

TMPM376FDDFG/FDFG contains 8-channel of TMRB.

Each channel functions independently and the channels operate in the same way except for the differences in their specification as shown in Table 8-1.

Table 8-1 Differences in the Specifications of TMRB Modules

Specification / Channel	External pins		Interrupt		Internal connection	
	External clock / capture trigger input pins	Timer flip-flop output pin	Capture interrupt	TMRB interrupt	ADC conversion start	Timer flip-flop output TBxOUT from SIO/UART (TXTRG: Transfer Clock)
	Signal name	Signal name				
TMRB0	TB0IN	TB0OUT	INTCAP00 INTCAP01	INTTB00 INTTB01		
TMRB1	TB1IN	TB1OUT	INTCAP10 INTCAP11	INTTB10 INTTB11		
TMRB2	TB2IN	TB2OUT	INTCAP20 INTCAP21	INTTB20 INTTB21		
TMRB3	TB3IN	TB3OUT	INTCAP30 INTCAP31	INTTB30 INTTB31		
TMRB4	TB4IN	TB4OUT	INTCAP40 INTCAP41	INTTB40 INTTB41		SIO0,SIO1
TMRB5	TB5IN	TB5OUT	INTCAP50 INTCAP51	INTTB50 INTTB51	INTTB51	
TMRB6	TB6IN	TB6OUT	INTCAP60 INTCAP61	INTTB60 INTTB61		
TMRB7	TB7IN	TB7OUT	INTCAP70 INTCAP71	INTTB70 INTTB71		SIO2,SIO3

8.3 Configuration

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

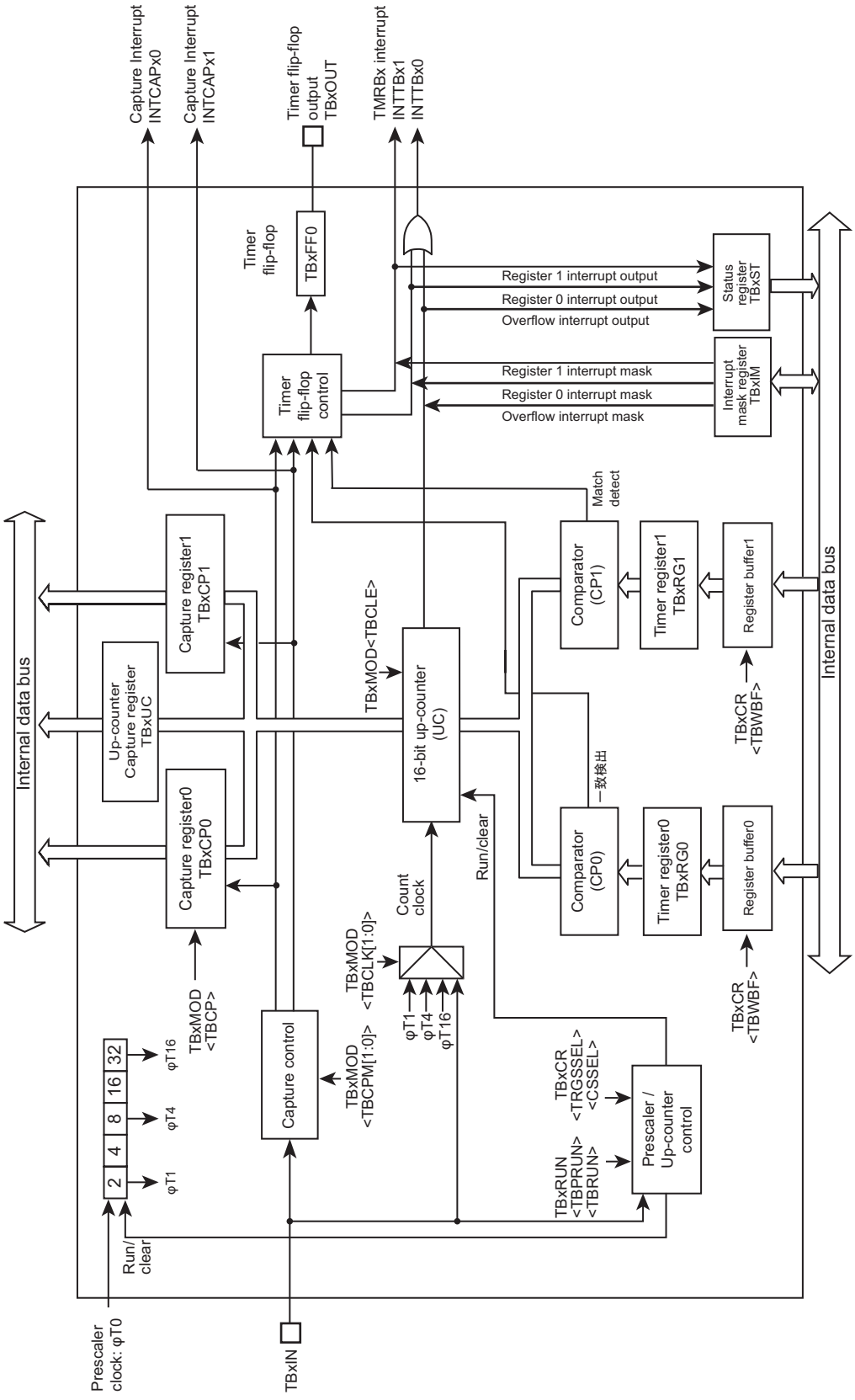


Figure 8-1 TMRBx Block Diagram (x= 0 to 7)

8.4 Registers

8.4.1 Register list according to channel

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

Channel x	Base Address
Channel0	0x4001_0000
Channel1	0x4001_0040
Channel2	0x4001_0080
Channel3	0x4001_00C0
Channel4	0x4001_0100
Channel5	0x4001_0140
Channel6	0x4001_0180
Channel7	0x4001_01C0

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

8.4.2 TBxEN(Enable register)

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

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

8.4.3 TBxRUN(RUN register)

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

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

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

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

8.4.4 TBxCR(Control register)

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

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

Note 1: Do not modify TBxCR during operating TMRB.

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

8.4.5 TBxMOD(Mode register)

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

Bit	Bit Symbol	Type	Function
31-7	–	R	Read as "0".
6	TBRSWR	R/W	Writes to timer registers 0 and 1 (when double buffering is enabled) 0: The data transfer to the timer register 0 and 1 is done by corresponding to the up-counter (UC) regardless of the rewriting of the buffer register 0 and 1. 1: To transfer the buffer registers data to the timer registers, the writing of the timer register 0 and 1 together are needed.
5	TBCP	W	Capture control by software 0: Capture by software 1: Don't care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as "1".
4-3	TBCPM[1:0]	R/W	Capture timing 00: Disable Capture timing 01: TBxIN↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. 10: TBxIN↑ TBxIN↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN pin input. 11: Disable Capture timing
2	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up counter when there is a match with Timer Regsiter1 (TBxRG1).
1-0	TBCLK[1:0]	R/W	Selects the TMRBx source clock. 00: TBxIN pin input 01: ϕT1 10: ϕT4 11: ϕT16

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

8.4.6 TBxFFCR(Flip-flop control register)

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

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

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

8.4.7 TBxST(Status register)

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

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

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

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

8.4.8 TBxIM(Interrupt mask register)

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

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

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

8.4.9 TBxUC(Up counter capture register)

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

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

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

8.4.10 TBxRG0(Timer register 0)

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

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

8.4.11 TBxRG1(Timer register 1)

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

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

8.4.12 TBxCP0(Capture register 0)

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

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

8.4.13 TBxCP1(Capture register 1)

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

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

8.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 8-1.

8.5.1 Prescaler

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

The prescaler input clock ϕT0 is fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCK[2:0]> in the CG. The peripheral clock, fperiph, is either fgear, a clock selected by CGSYSCR<FPSEL> in the CG, or fc, which is a clock before it is divided by the clock gear.

The operation or the stoppage of a prescaler is set with TBxRUN<TBPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 8-2 show prescaler output clock resolutions.

Table 8-2 Prescaler Output Clock Resolutions (fc = 80MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			ϕT1	ϕT4	ϕT16
0 (fgear)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025 μs)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)
		001 (fperiph/2)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		010 (fperiph/4)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		011 (fperiph/8)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		100 (fperiph/16)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		101 (fperiph/32)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
	101 (fc/4)	000 (fperiph/1)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		001 (fperiph/2)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		010 (fperiph/4)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		011 (fperiph/8)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		100 (fperiph/16)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		101 (fperiph/32)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
	111 (fc/16)	000 (fperiph/1)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		001 (fperiph/2)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		010 (fperiph/4)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		011 (fperiph/8)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
		100 (fperiph/16)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)	$fc/2^{13}$ (102.4 μs)
		101 (fperiph/32)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)	$fc/2^{14}$ (204.8 μs)

Table 8-2 Prescaler Output Clock Resolutions (fc = 80MHz)

Select peripheral clock CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Select prescaler clock CGSYSCR <PRCK[2:0]>	Prescaler output clock function		
			ϕT1	ϕT4	ϕT16
1 (fc)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.025 μs)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)
		001 (fperiph/2)	fc/2^2 (0.05 μs)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
	100 (fc/2)	000 (fperiph/1)	–	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)
		001 (fperiph/2)	fc/2^2 (0.05 μs)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
	101 (fc/4)	000 (fperiph/1)	–	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)
		001 (fperiph/2)	–	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	fc/2^5 (0.4 μs)
		001 (fperiph/2)	–	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)
		010 (fperiph/4)	–	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	fc/2^5 (0.4 μs)
		001 (fperiph/2)	–	–	fc/2^6 (0.8 μs)
		010 (fperiph/4)	–	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		011 (fperiph/8)	–	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)

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

8.5.2 Up-counter (UC)

UC is a 16-bit binary counter.

- Source clock

  UC source clock, specified by TBxMOD<TBCLK[1:0]>, can be selected from either three types ϕT1, ϕT4, ϕT16 of prescaler output clock or the external clock of the TBxIN pin.

- Count start / stop

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

- Timing to clear UC

  1. When a match is detected

     By setting TBxMOD<TBCLE> = "1", UC is cleared if when the comparator detects a match between counter value and the value set in TBxRG1. UC operates as a free-running counter if TBxMOD<TBCLE> = "0".

  2. When UC stops

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

- UC overflow

  If UC overflow occurs, the INTTBx0 overflow interrupt is generated.

8.5.3 Timer registers (TBxRG0, TBxRG1)

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

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

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

8.5.4 Capture

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

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

8.5.5 Capture register (TBxCP0, TBxCP1)

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

8.5.6 Up counter capture register (TBxUC)

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

8.5.7 Comparators (CP0, CP1)

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

8.5.8 Timer Flip-flop (TBxFF0)

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

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

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

8.5.9 Capture interrupt (INTCAPx0, INTCAPx1)

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

8.6 Description of Operations for Each Mode

8.6.1 16-bit Interval Timer Mode

In the case of generating constant period interrupt, set the interval time to the Timer register (TBxRG0) to generate the INTTBx0 interrupt. Same as TBxRG0, INTTBx1 interrupt is generated by setting different interval time value to TBxRG1 timer resister.

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx1 interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger.
TBxMOD	←	X	0	1	0	0	1	*	*	Changes to prescaler output clock as input clock. Specifies capture function to disable. (** = 01, 10, 11)
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don't care
–; No change

8.6.2 16-bit Event Counter Mode

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

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

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

Note: X; Don't care
–; No change

8.6.3 16-bit PPG (Programmable Pulse Generation) Output Mode

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

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

Set value of TBxRG0 < Set value of TBxRG1

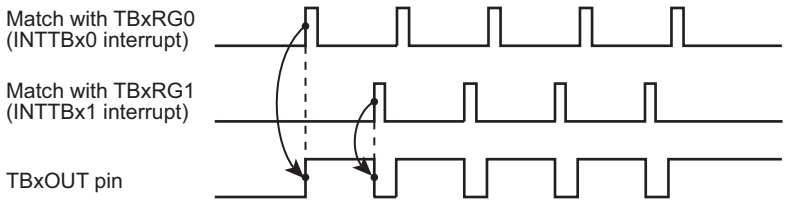


Figure 8-2 Example of Output of Programmable Pulse Generation (PPG)

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

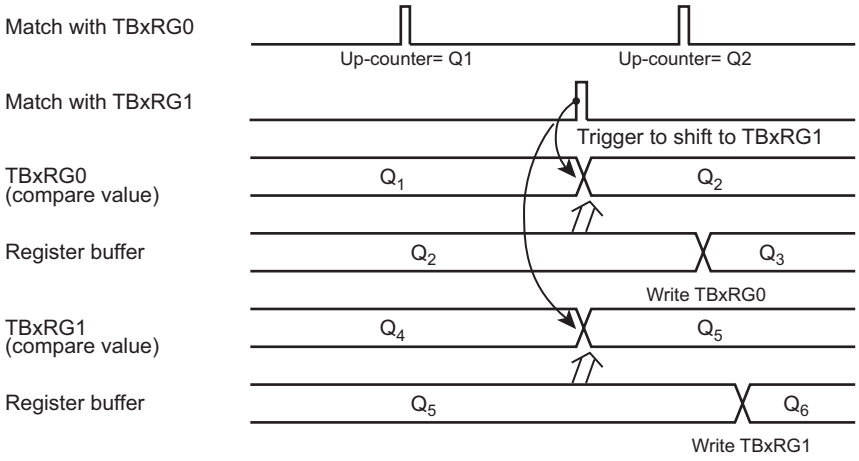


Figure 8-3 Register Buffer Operation

The block diagram of this mode is shown below.

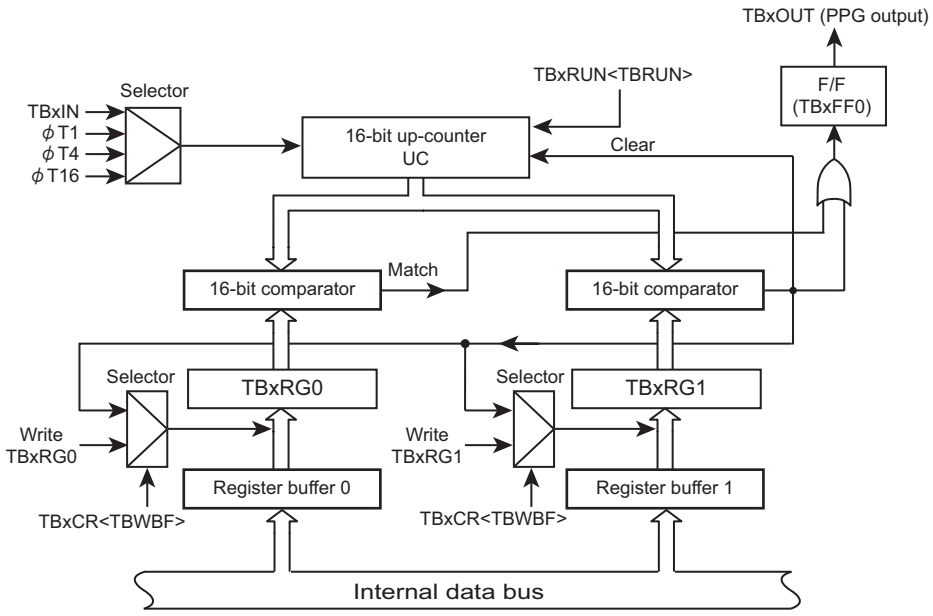


Figure 8-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	–	X	–	X	0	0	Disables double buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	0	X	–	X	0	0	Enables the TBxRG0 double buffering.
										(Changes the duty/cycle when the INTTBx0 interrupt is generated)
TBxFFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TBxFF0 to reverse when a match with TBxRG0 or TBxRG1 is detected, and sets the initial value of TBxFF0 to "0".
TBxMOD	←	X	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function.
							(** = 01, 10, 11)			UC is cleared to match TBxRG1.
Set PORT registers.										Allocates corresponding port to TBxOUT.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don't care
–; No change

8.6.4 External trigger Programmable Pulse Generation Output Mode (PPG)

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

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

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

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

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

Symbols (d) and (p) used in the text correspond to symbols d and p in Figure 8-5.

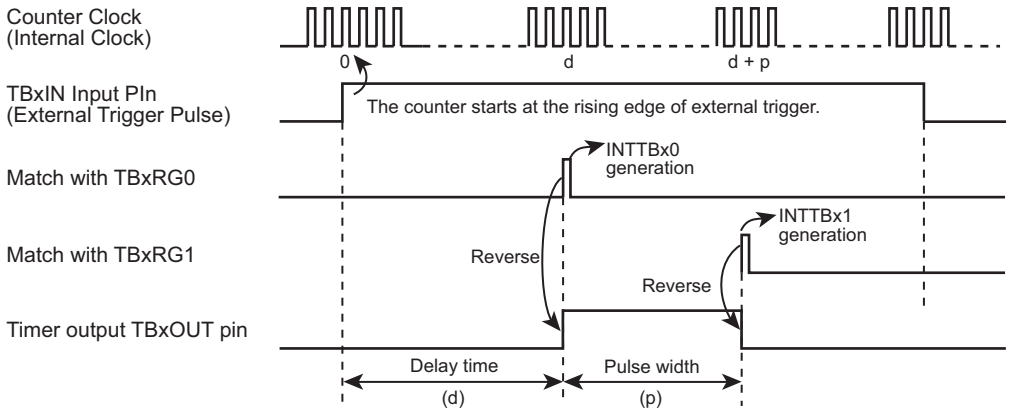


Figure 8-5 One-shot pulse generation using an external count start trigger (with a delay)

8.7 Applications using the Capture Function

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

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

8.7.1 One-shot pulse output triggered by an external pulse

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

The 16-bit up-counter is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TBxIN pin. A trigger is generated at the rising of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TBxCP0).

The CPU must be programmed so that an interrupt INTCAPx0 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TBxRG0) to the sum of the TBxCP0 value (c) and the delay time (d), (c + d), and set the timer registers (TBxRG1) to the sum of the TBxRG0 values and the pulse width (p) of one-shot pulse, (c + d + p).[TBxRG1 change must be completed before the next match.]

In addition, the timer flip-flop control registers(TBxFFCR<TBE1T1, TBE0T1>) must be set to "11". This enables triggering the timer flip-flop (TBxFF0) to reverse when TBxUC matches TBxRG0 and TBxRG1. This trigger is disabled by the INTTBx0 / INTTBx1 interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in Figure 8-6.

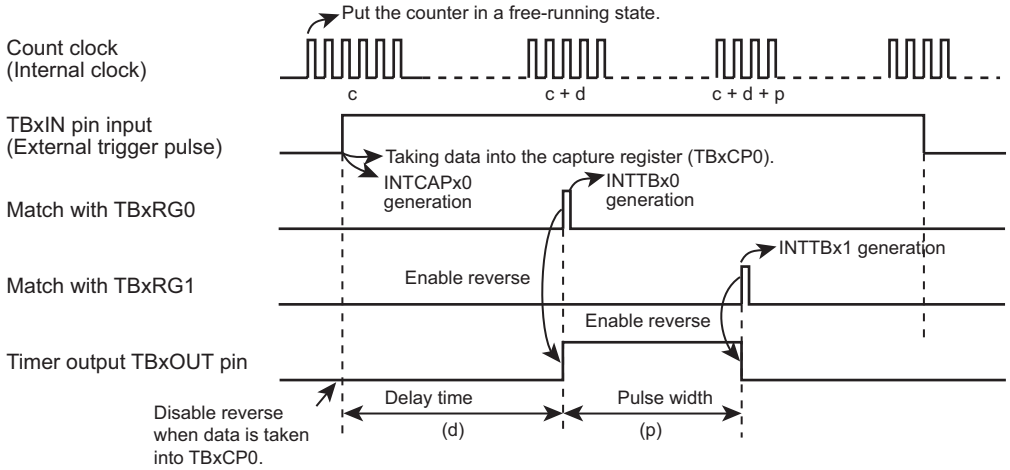


Figure 8-6 One-shot Pulse Output (With Delay)

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

		7	6	5	4	3	2	1	0	
[[Main processing] Capture setting by TBxIN										
Set PORT registers.										Allocates corresponding port to TBxIN.
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxMOD	←	X	0	1	0	1	0	0	1	Changes source clock to ΦT1. Fetches a count value into the TBxCP0 at the rising edge of TBxIN.
TBxFFCR	←	X	X	0	0	0	0	1	0	Clears TBxFF0 reverse trigger and disables.
Set PORT registers.										Allocates corresponding port to TBxOUT.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupts specified by INTCAPx0 interrupt corresponding bit by setting to "1".
TBxRUN	←	*	*	*	*	*	1	X	1	Starts the TMRBx module.
[Processing of INTCAPx0 interrupt service routine] Pulse output setting										
TBxRG0	←	*	*	*	*	*	*	*	*	Sets count value.(TBxCAP0 + 3ms/ΦT1)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Sets count value.(TBxCAP0 + (3+2)ms/ΦT1)
	←	*	*	*	*	*	*	*	*	
TBxFFCR	←	X	X	–	–	1	1	–	–	Reverses TBxFF0 if UC consistent with TBxRG0 and TBxRG1.
TBxIM	←	X	X	X	X	X	1	0	1	Masks except TBxRG1 correspondence interrupt.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupt specified by INTTBx interrupt corresponding bit setting to "1".
[Processing of INTTBx interrupt service routine] Output disable										
TBxFFCR	←	X	X	–	–	0	0	–	–	Clears TBxFF0 reverse trigger setting.
	←	*	*	*	*	*	*	*	*	Prohibits interrupts specified by INTTBx interrupt corresponding bit by setting to "1".

Note: X; Don't care
–; No change

If a delay is not required, TBxFF0 is reversed when data is taken into TBxCP0, and TBxRG1 is set to the sum of the TBxCP0 value (c) and the one-shot pulse width (p), (c + p), by generating the INTCAPx0 interrupt. TBxRG1 change must be completed before the next match.

TBxFF0 is enabled to reverse when UC matches with TBxRG1, and is disabled by generating the INTTBx1 interrupt.

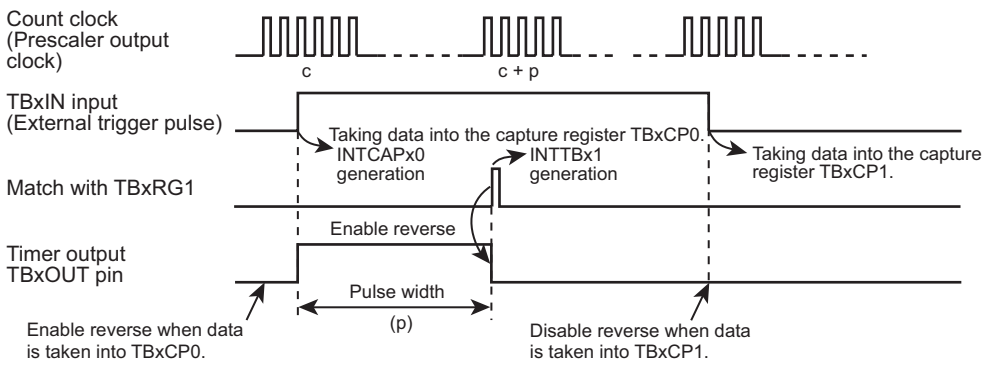


Figure 8-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

8.7.2 Pulse width measurement

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

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

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

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

The "Low" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAPx0 interrupt processing as shown in Figure 8-8 and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

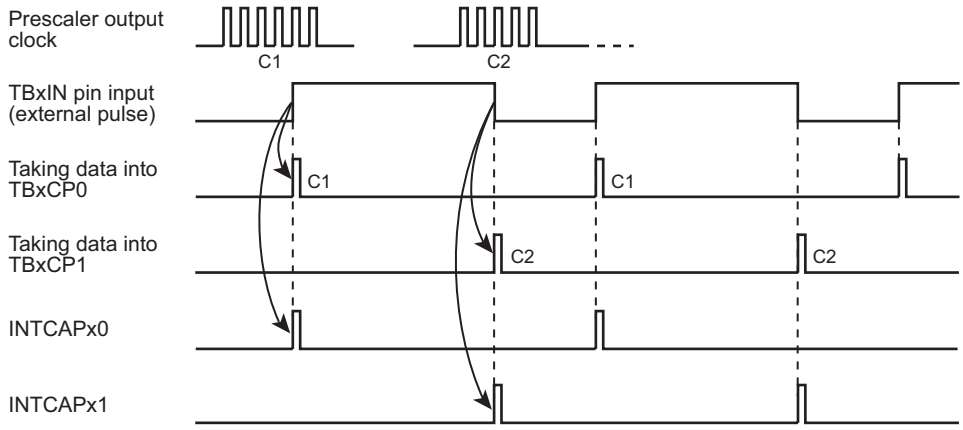


Figure 8-8 Pulse Width Measurement

9. Serial Channel (SIO/UART)

9.1 Overview

This device has two mode for the serial channel, one is the synchronous communication mode (I/O interface mode), and the other is the asynchronous communication mode (UART mode).

Their features are given in the following.

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

  The usable double buffer function, and the usable FIFO buffers of transmit and receive in all for maximum 4-byte.
- I/O Interface Mode
  - Transfer Mode: the half duplex (transmit/receive), the full duplex
  - Clock: Output (fixed rising edge) /Input (selectable rising/falling edge)
  - Make it possible to specify the interval time of continuous transmission.
- UART Mode
  - Data length: 7 bits, 8bits, 9bits
  - Add parity bit (to be against 9bits data length)
  - Serial links to use wake-up function
  - Handshaking function with $\overline{\text{CTS}}$ pin

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

9.2 Difference in the Specifications of SIO Modules

TMPM376FDDFG/FDFG has four SIO channels.

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

Table 9-1 Difference in the Specifications of SIO Modules

	Pin name			Interrupt		UART source clock
	TXD	RXD	$\overline{\text{CTS}}$x/ SCLKx	Receive Interrupt	Transmit Interrupt	
Channel 0	PE0	PE1	PE2	INTRX0	INTTX0	TB4OUT
Channel 1	PA5	PA6	PA4	INTRX1	INTTX1	TB4OUT
Channel 2	PD5	PD6	PD4	INTRX2	INTTX2	TB7OUT
Channel 3	PF3	PF4	PF2	INTRX3	INTTX3	TB7OUT

9.3 Configuration

Figure 9-1 shows SIO block diagram.

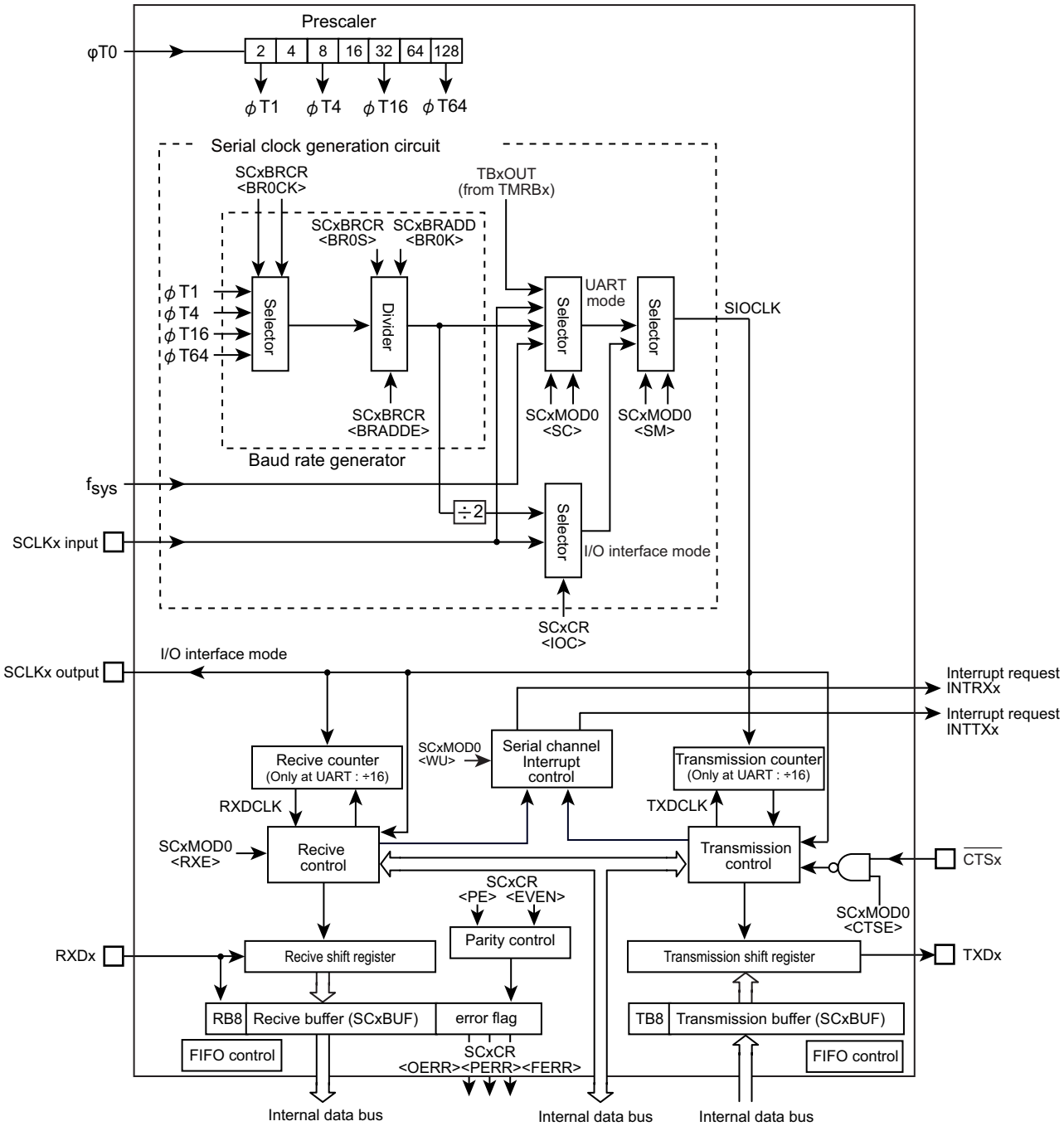


Figure 9-1 SIO Block Diagram

9.4 Registers Description

9.4.1 Registers List in Each Channel

The each channel registers and addresses are shown below.

Channel x	Base Address
Channel0	0x4002_0080
Channel1	0x4002_00C0
Channel2	0x4002_0100
Channel3	0x4002_0140

Register name (x=0,1,2,3)		Address (Base+)
Enable register	SCxEN	0x0000
Buffer register	SCxBUF	0x0004
Control register	SCxCR	0x0008
Mode control register 0	SCxMOD0	0x000C
Baud rate generator control register	SCxBRCR	0x0010
Baud rate generator control register 2	SCxBRADD	0x0014
Mode control register 1	SCxMOD1	0x0018
Mode control register 2	SCxMOD2	0x001C
RX FIFO configuration register	SCxRFC	0x0020
TX FIFO configuration register	SCxTFC	0x0024
RX FIFO status register	SCxRST	0x0028
TX FIFO status register	SCxTST	0x002C
FIFO configuration register	SCxFCNF	0x0030

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

9.4.2 SCxEN (Enable Register)

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

Bit	Bit Symbol	Type	Function
31-1	–	R	Read as 0.
0	SIOE	R/W	SIO operation 0: Disabled 1: Enabled Specified the SIO operation. To use the SIO, set <SIOE> = "1". When the operation is disabled, no clock is supplied to the other registers in the SIO module. This can reduce the power consumption. If the SIO operation is executed and then disabled, the settings will be maintained in each register except for SCxTFC<TIL[1:0]>.

Note: When SCxEN<SIOE> is cleared to "0" (disable SIO operation) or the operation mode transits to IDLE mode by setting SCxMOD1<I2S0> to "0", it is necessary to reset SCxTFC.

9.4.3 SCxBUF (Buffer Register)

SCxBUF works as a transmit buffer or FIFO for write operation and as a receive buffer or FIFO for read operation.

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-0	TB[7:0] / RB[7:0]	R/W	[write] TB : Transmit buffer / FIFO [read] RB : Receive buffer / FIFO

9.4.4 SCxCR (Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	RB8	R	Receive data bit 8 (For UART) 9th bit of the received data in the 9 bits UART mode.
6	EVEN	R/W	Parity (For UART) 0: Odd 1: Even Selects even or odd parity. "0" : odd parity, "1" : even parity. The parity bit may be used only in the 7- or 8-bit UART mode.
5	PE	R/W	Add parity (For UART) 0: Disabled 1: Enabled Controls enabling/ disabling parity. The parity bit may be used only in the 7- or 8-bit UART mode.
4	OERR	R	Overrun error flag (Note) 0: Normal operation 1: Error
3	PERR	R	Parity / Under-run error flag (Note) 0: Normal operation 1: Error
2	FERR	R	Framing error flag (Note) 0: Normal operation 1: Error
1	SCLKS	R/W	Selects input clock edge for data transmission and reception. (For I/O Interface) 0: Data in the transmit buffer is sent to TXDx pin one bit at a time on the falling edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the rising edge of SCLKx. In this case, the SCLK starts from high level. 1: Data in the transmit buffer is sent to TXDx pin one bit at a time on the rising edge of SCLKx. Data from RXDx pin is received in the receive buffer one bit at a time on the falling edge of SCLKx. In this case, the SCLK starts from low level. Set to "0" in the clock output mode.
0	IOC	R/W	Selecting clock (For I/O Interface) 0: Baud rate generator 1: SCLK pin input

Note: **Any error flag (OERR, PERR, FERR) is cleared to "0" when read.**

9.4.5 SCxMOD0 (Mode Control Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TB8	CTSE	RXE	WU	SM		SC	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	TB8	R/W	Transmit data bit 8 (For UART) Writes the 9th bit of transmit data in the 9 bits UART mode.
6	CTSE	R/W	Handshake function control (For UART) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{CTS}$ pin.
5	RXE	R/W	Receive control (Note1)(Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For UART) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. In it is Enabled, Interrupt only when RB9 = "1" at 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For UART) 00: Timer TBxOUT (Refer to Table 9-1) 01: Baud rate generator 10: Internal clock fsys 11: External clock (SCLK input) (As for the I/O interface mode, the serial transfer clock can be set in the control register (SCxCR).

Note 1: With <RXE> set to "0", set each mode register (SCxMOD0, SCxMOD1 and SCxMOD2). Then set <RXE> to "1".

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

9.4.6 SCxMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	I2S0	FDPX		TXE	SINT			-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	I2S0	R/W	IDLE 0: Stop 1: Operate Specifies the IDLE mode operation.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.
4	TXE	R/W	Transmit control (Note1)(Note2) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface) 000: None 001: 1SCLK 010: 2SCLK 011: 4SCLK 100: 8SCLK 101: 16SCLK 110: 32SCLK 111: 64SCLK This parameter is valid only for the I/O interface mode when SCLK pin output is selected. In other modes, this function has no meaning. Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.
0	–	R/W	Write a "0".

Note 1: **Specify the all mode first and then enable the <TXE> bit.**

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

Note 3: When SCxEN<SIOE> is cleared to "0" (disable SIO operation) or the operation mode transits to IDLE mode by setting SCxMOD1<I2S0> to "0", it is necessary to reset SCxTFC.

9.4.7 SCxMOD2 (Mode Control Register 2)

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

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>–</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7</td><td>TBEMP</td><td>R</td><td>Transmit buffer empty flag.
0: Full
1: Empty
If double buffering is disabled, this flag is insignificant.
This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1".
Writing data again to the double buffers sets this bit to "0".</td></tr>
<tr><td>6</td><td>RBFLL</td><td>R</td><td>Receive buffer full flag.
0: Empty
1: Full
If double buffering is disabled, this flag is insignificant.
This is a flag to show that the receive double buffers are full.
When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0".</td></tr>
<tr><td>5</td><td>TXRUN</td><td>R</td><td>In transmission flag
0: Stop
1: Operate
This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> bits indicate the following status.
<table>
<tr><th><TXRUN></th><th><TBEMP></th><th>Status</th></tr>
<tr><td>1</td><td>–</td><td>Transmission in progress</td></tr>
<tr><td rowspan="2">0</td><td>1</td><td>Transmission completed</td></tr>
<tr><td>0</td><td>Wait state with data in Transmit buffer</td></tr>
</table></td></tr>
<tr><td>4</td><td>SBLEN</td><td>R/W</td><td>STOP bit (for UART)
0 : 1-bit
1 : 2-bit
This specifies the length of transmission stop bit in the UART mode.
On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.</td></tr>
<tr><td>3</td><td>DRCHG</td><td>R/W</td><td>Setting transfer direction
0: LSB first
1: MSB first
Specifies the direction of data transfer in the I/O interface mode.
In the UART mode, set this bit to LSB first.</td></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Double-buffer
0: Disabled
1 : Enabled
This parameter enables or disables the transmit/receive double buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART mode.
When receiving data in the I/O interface mode (SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.</td></tr>
<tr><td>1-0</td><td>SWRST[1:0]</td><td>R/W</td><td>Software reset
Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the following bits are initialized :
<table>
<tr><th>Register</th><th>Bit</th></tr>
<tr><td>SCxMOD0</td><td><RXE></td></tr>
<tr><td>SCxMOD1</td><td><TXE></td></tr>
<tr><td>SCxMOD2</td><td><TBEMP>, <RBFLL>, <TXRUN></td></tr>
<tr><td>SCxCR</td><td><OERR>, <PERR>, <FERR></td></tr>
</table>
The transmit/receive circuit and the FIFO become initial state (see Note1 and Note2).</td></tr>
</table>

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

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

9.4.8 SCxBRCR (Baud Rate Generator Control Register), SCxBRADD (Baud Rate Generator Control Register 2)

The division ratio of the baud rate generator can be specified in the registers shown below.

SCxBRCR

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	BRADDE	BR0CK		BR0S			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	–	R/W	Write "0".
6	BRADDE	R/W	N + (16 – K)/16 divider function (For UART) 0: disabled 1: enabled This division function can only be used in the UART mode.
5-4	BR0CK[1:0]	R/W	Select input clock to the baud rate generator 00: ϕT1 01: ϕT4 10: ϕT16 11: ϕT64
3-0	BR0S[3:0]	R/W	Division ratio "N" 0000: 16 0001: 1 0010: 2 ... 1111: 15

SCxBRADD

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	BR0K			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-0	BR0K[3:0]	R/W	Specify K for the "N + (16 − K)/16" division (For UART) 0000: Prohibited 0001: K = 1 0010: K = 2 ... 1111: K = 15

Table 9-2 lists the settings of baud rate generator division ratio.

Table 9-2 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only UART mode)
<BR0S>	Specify "N" (Note2) (Note3)	
<BR0K>	No setting required	Specify "K" (Note4)
Division ratio	Divide by N	$N + \frac{(16 - K)}{16}$ division.

Note 1: **To use the "N + (16 - K)/16" division function, be sure to set <BRADDE> to "1" after setting the K value to <BR0K>. The "N + (16 - K)/16" division function can only be used in the UART mode.**

Note 2: **As a division ratio, 1 ("0001") or 16 ("0000") can not be applied to N when using the "N + (16 - K)/16" division function in the UART mode.**

Note 3: **The division ratio "1" of the baud rate generator can be specified only when the double buffering is used in the I/O interface mode.**

Note 4: **Specifying "K = 0" is prohibited.**

9.4.9 SCxFCNF (FIFO Configuration Register)

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

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

Note 1: **Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.**

Note 2: **The FIFO can not use in 9bit UART mode.**

9.4.10 SCxRFC (RX FIFO Configuration Register)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	RFCS	W	RX FIFO clear (Note) 1: Clears RX FIFO When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL> is "000". And also the read pointer is initialized.
6	RFIS	R/W	Select 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 exceeds the specified fill level at the time data is read.
5-2	–	R	Read as 0.
1-0	RIL[1:0]	R/W	FIFO fill level to generate RX interrupts

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

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

9.4.11 SCxTFC (TX FIFO Configuration Register) (Note2)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	TFCS	W	TX FIFO clear (Note 1) 1: Clears TX FIFO. When SCxTST<TFCS> is set to "1", the transmit FIFO is cleared and SCxRST<TLVL> is "000". And also the write pointer is initialized.
6	TFIS	R/W	Selects interrupt generation condition. 0: An interrupt is generated when the data reaches to the specified fill level. 1: An interrupt is generated when the data reaches to the specified fill level or the data can not reach the specified fill level at the time new data is read.
5-2	–	R	Read as 0.
1-0	TIL[1:0]	R/W	FIFO fill level to generate TX interrupts.

	Other than full duplex	Full duplex
00	Empty	Empty
01	1 byte	1 byte
10	2 byte	Empty
11	3 byte	1 byte

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

Note 2: **After you perform the following operations, configure the SCxTFC register again.**
SCxEN<SIOE> = "0" (SIO operation stop)
Conditions are as follows:SCxMOD1<I2S0> = "0" (operation is prohibited in IDLE mode) and releasing the low power consumption mode which started by the WFI (Wait For Interrupt) instruction.

9.4.12 SCxRST (RX FIFO Status Register)

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

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

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

9.4.13 SCxTST (TX FIFO Status Register)

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

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

Note: **The <TUR> bit is cleared to "0" when transmit data is written to the SCxBUF register.**

9.5 Operation in Each Mode

Table 9-3 shows the modes and data formats.

Table 9-3 Mode and Data format

Mode	Mode type	Data length	Transfer direction	Specifies whether to use parity bits.	STOP bit length (transmit)
Mode 0	Synchronous communication mode (IO interface mode)	8 bit	LSB first/MSB first	-	-
Mode 1	Asynchronous communication mode (UART mode)	7 bit	LSB first	o	1 bit or 2 bit
Mode 2		8 bit		o	
Mode 3		9 bit		×	

Mode 0 is a synchronous communication and can be used to extend I/O. This mode transmits and receives data in synchronization with SCLK. SCLK can be used for both input and output.

The direction of data transfer can be selected from LSB first and MSB first. This mode is not allowed either to use parity bits or STOP bits.

The mode 1, mode 2 and mode 3 are asynchronous modes and the transfer direction is fixed to the LSB first.

Parity bits can be added in the mode 1 and mode 2. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system).

STOP bit in transmission can be selected from 1 bit and 2 bits. The STOP bit length in reception is fixed to a one bit.

9.6 Data Format

9.6.1 Data Format List

Figure 9-2 shows data format.

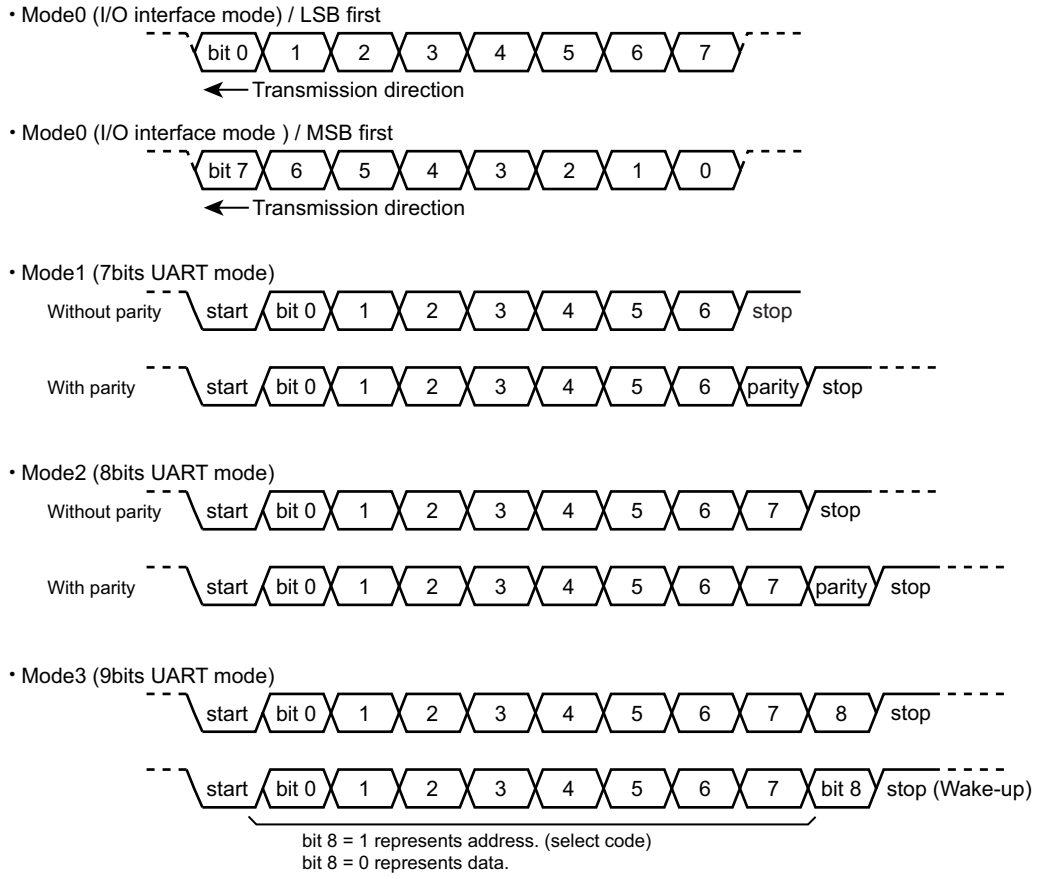


Figure 9-2 Data Format

9.6.2 Parity Control

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

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

The <EVEN> bit of SCxCR selects either even or odd parity.

9.6.2.1 Transmission

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

After data transmission is complete, the parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode and SCxMOD0<TB8> in the 8-bit UART mode.

The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

9.6.2.2 Receiving Data

If the received data is moved from the receive shift register to the receive buffer, a parity is generated.

In the 7-bit UART mode, the generated parity is compared with the parity stored in SCxBUF<RB7>, while in the 8-bit UART mode, it is compared with the one in SCxCR<RB8>.

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

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

9.6.3 STOP Bit Length

The length of the STOP bit in the UART transmission mode can be selected from one bit or two bits by setting the SCxMOD2<SBLEN>. The length of the STOP bit data is determined as one-bit when it is received regardless of the setting of this bit.

9.7 Clock Control

9.7.1 Prescaler

There is a 7-bit prescaler to divide a prescaler input clock ΦT0 by 2, 8, 32 and 128.

Use the CGSYSCR register in the clock/mode control block to select the input clock ΦT0 of the prescaler.

The prescaler becomes active only when the baud rate generator is selected as a transfer clock by SCxMOD0<SC[1:0]> = "01".

Table 9-4 show the resolution of the input clock to the baud rate generator.

Table 9-4 Clock Resolution to the Baud Rate Generator fc = 80 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			ϕT1	ϕT4	ϕT16	ϕT64
0 (fgear)	000 (fc)	000 (fperiph/1)	fc/2^1 (0.025 μs)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)
		001 (fperiph/2)	fc/2^2 (0.05 μs)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		010 (fperiph/4)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		011 (fperiph/8)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		100 (fperiph/16)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		101 (fperiph/32)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
	100 (fc/2)	000 (fperiph/1)	fc/2^2 (0.05 μs)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)
		001 (fperiph/2)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		010 (fperiph/4)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		011 (fperiph/8)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		100 (fperiph/16)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		101 (fperiph/32)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
	101 (fc/4)	000 (fperiph/1)	fc/2^3 (0.1 μs)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)
		001 (fperiph/2)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		010 (fperiph/4)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		011 (fperiph/8)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		100 (fperiph/16)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
		101 (fperiph/32)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)	fc/2^{14} (204.8 μs)
	110 (fc/8)	000 (fperiph/1)	fc/2^4 (0.2 μs)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)
		001 (fperiph/2)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		010 (fperiph/4)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		011 (fperiph/8)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
		100 (fperiph/16)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)	fc/2^{14} (204.8 μs)
		101 (fperiph/32)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)	fc/2^{15} (409.6 μs)
	111 (fc/16)	000 (fperiph/1)	fc/2^5 (0.4 μs)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)
		001 (fperiph/2)	fc/2^6 (0.8 μs)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)
		010 (fperiph/4)	fc/2^7 (1.6 μs)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)
		011 (fperiph/8)	fc/2^8 (3.2 μs)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)	fc/2^{14} (204.8 μs)
		100 (fperiph/16)	fc/2^9 (6.4 μs)	fc/2^{11} (25.6 μs)	fc/2^{13} (102.4 μs)	fc/2^{15} (409.6 μs)
		101 (fperiph/32)	fc/2^{10} (12.8 μs)	fc/2^{12} (51.2 μs)	fc/2^{14} (204.8 μs)	fc/2^{16} (819.2 μs)

Table 9-4 Clock Resolution to the Baud Rate Generator fc = 80 MHz

peripheral clock selection CGSYSCR <FPSEL>	Clock gear value CGSYSCR <GEAR[2:0]>	Prescaler clock selection CGSYSCR <PRCK[2:0]>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
1 (fc)	000 (fc)	000 (fperiph/1)	$fc/2^1$ (0.025 μs)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	100 (fc/2)	000 (fperiph/1)	–	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	$fc/2^2$ (0.05 μs)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	101 (fc/4)	000 (fperiph/1)	–	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	–	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	$fc/2^3$ (0.1 μs)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	110 (fc/8)	000 (fperiph/1)	–	–	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	–	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	–	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	$fc/2^4$ (0.2 μs)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)
	111 (fc/16)	000 (fperiph/1)	–	–	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)
		001 (fperiph/2)	–	–	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)
		010 (fperiph/4)	–	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)
		011 (fperiph/8)	–	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)
		100 (fperiph/16)	$fc/2^5$ (0.4 μs)	$fc/2^7$ (1.6 μs)	$fc/2^9$ (6.4 μs)	$fc/2^{11}$ (25.6 μs)
		101 (fperiph/32)	$fc/2^6$ (0.8 μs)	$fc/2^8$ (3.2 μs)	$fc/2^{10}$ (12.8 μs)	$fc/2^{12}$ (51.2 μs)

Note 1: **The prescaler output clock φTn must be selected so that the relationship "φTn ≤ fsys / 2" is satisfied (so that φTn is slower than fsys).**

Note 2: **Do not change the clock gear while SIO is operating.**

Note 3: **The dashes in the above table indicate that the setting is prohibited.**

9.7.2 Serial Clock Generation Circuit

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

9.7.2.1 Baud Rate Generator

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

(1) Baud Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 2, 8, 32 and 128.

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

(2) Baud Rate Generator output clock

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

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

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

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BR0S>	Divide by K SCxBRADD<BR0K>
I/O interface	Divide by N	1 to 16 (Note)	-
UART	Divide by N	1 to 16	-
	N + (16-K)/16 division	2 to 15	1 to 15

Note: 1/N (N=1)frequency division ratio can be used only when a double buffer is enabled.

9.7.2.2 Clock Selection Circuit

A clock can be selected by setting the modes and the register.

Modes can be specified by setting the SCxMOD0<SM>.

The input clock in I/O interface mode is selected by setting SCxCR. The clock in UART mode is selected by setting SCxMOD0<SC>.

(1) Transfer Clock in I/O interface mode

Table 9-5 shows clock selection in I/O interface mode.

Table 9-5 Clock Selection in I/O Interface Mode

Mode SCxMOD0<SM>	Input/Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
I/O interface mode	SCLK output	Set to "0". (Fixed to the rising edge)	Divided by 2 of the baud rate generator output.
	SCLK input	Rising edge	SCLK input rising edge
		Falling edge	SCLK input falling edge

To get the highest baud rate, the baud rate generator must be set as below.

Note: **When deciding clock settings, make sure that AC electrical character is satisfied.**

- Clock/mode control block settings
  - fc = 80MHz
  - fgear = 80MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - ϕT0 = 80MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)

- SIO settings (if double buffer is used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : ϕT1 selected) = 40MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0001" : 1 division ratio) = 40MHz

  1 division ratio can be selected if double buffer is used. In this case, baud rate is 20Mbps because 40MHz is divided by 2.

- SIO settings (if double buffer is not used)
  - Clock (SCxBRCR<BRCK[1:0]> = "00" : ϕT1 selected) = 40MHz
  - Divided clock frequency (SCxBRCR<BRS[3:0]> = "0010" : 2 division ratio) = 20MHz

  2 division ratio is the highest if double buffer is not used. In this case, baud rate is 10Mbps because 20MHz is divided by 2.

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

- If double buffer is used
  - SCLK cycle > 6/fsys

  The highest baud rate is less than 80 ÷ 6 = 13.3 Mbps.

- If double buffer is not used

  - SCLK cycle > 8/fsys

  The highest baud rate is less than 80 ÷ 8 = 10 Mbps.

(2) Transfer clock in the UART mode

Table 9-6 shows the clock selection in the UART mode. In the UART mode, selected clock is divided by 16 in the receive counter or the transmit counter before use.

Table 9-6 Clock Selection in UART Mode

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

The examples of baud rate in each clock settings.

- If the baud rate generator is used
  - fc = 80MHz
  - fgear = 80MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
  - ϕT0 = 80MHz (CGSYSCR<PRCK[2:0]> = "000" : 1 division ratio)
  - Clock = ΦT1 = 40MHz (SCxBRCR<BRCK[1:0]> = "00" : ϕT1 selected)

  The highest baud rate is 2.5Mbps because 40MHz is divided by 16.

Table 9-7 shows examples of baud rate when the baud rate generator is used with the following clock settings.

- fc = 9.8304MHz
- fgear = 9.8304MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
- ϕT0 = 4.9152MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)

Table 9-7 Example of UART Mode Baud Rate (Using the Baud Rate Generator)

fc [MHz]	Division ratio N (SCxBRCR<BRS[3:0]>)	ϕT1 (fc/4)	ϕT4 (fc/16)	ϕT16 (fc/64)	ϕT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	16	9.600	2.400	0.600	0.150

Unit：kbps

- If the SCLK input is used

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

- SCLK cycle > 2/fsys

The highest baud rate must be less than 80 ÷ 2 ÷ 16 = 2.5 Mbps.

- If fsys is used

Since the highest value of fsys is 80MHz, the highest baud rate is 80 ÷ 16 = 5Mbps.

- If timer output is used

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

Baud rates can be obtained by using the following formula.

Baud rate calculation

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

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

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

Table 9-8 shows the examples of baud rates when the timer output is used with the following clock settings.

- fc = 80MHz / 9.8304MHz / 8MHz
- fgear = 80MHz / 9.8304MHz / 8MHz (CGSYSCR<GEAR[2:0]> = "000" : fc selected)
- ϕT0 = 40MHz / 4.9152MHz / 4MHz (CGSYSCR<PRCK[2:0]> = "001" : 2 division ratio)
- Timer count clock = 4MHz / 1.2287MHz / 1MHz (TBxMOD<TBCLK[1:0]> = "01" : ϕT1 selected)

Table 9-8 Example of UART Mode Baud Rate (Using the Timer Output)

TBxRG0 setting	fc		
	80MHz	9.8304MHz	8MHz
0x0001	625	76.8	62.5
0x0002	312.5	38.4	31.25
0x0003	-	25.6	-
0x0004	156.25	19.2	15.625
0x0005	125	15.36	12.5
0x0006	-	12.8	-
0x0008	78.125	9.6	-
0x000A	62.5	7.68	6.25
0x0010	39.025	4.8	-
0x0014	31.25	3.84	3.125

Unit：kbps

9.8 Transmit/Receive Buffer and FIFO

9.8.1 Configuration

Figure 9-3 shows the configuration of transmit buffer, receive buffer and FIFO.

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

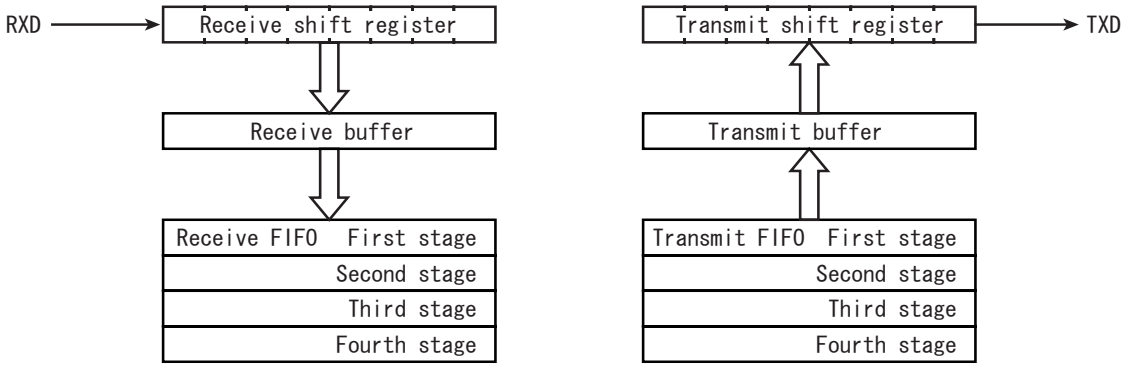


Figure 9-3 The Configuration of Buffer and FIFO

9.8.2 Transmit/Receive Buffer

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

In the case of using a receive buffer, if SCLK input is set to generate clock output in the I/O interface mode or the UART mode is selected, it's double buffered despite the <WBUF> settings. In other modes, it's according to the <WBUF> settings.

Table 9-9 shows correlation between modes and buffers.

Table 9-9 Mode and buffer Composition

Mode		SCxMOD2<WBUF>	
		"0"	"1"
UART	Transmit	Single	Double
	Receive	Double	Double
I/O interface (SCLK input)	Transmit	Single	Double
	Receive	Double	Double
I/O interface (SCLK output)	Transmit	Single	Double
	Receive	Single	Double

9.8.3 FIFO

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

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

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

Table 9-10 shows correlation between modes and FIFO.

Table 9-10 Mode and FIFO Composition

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

9.9 Status Flag

The SCxMOD2 register has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1" while reading this bit changes it to "0".

<TBEMP> shows that the transmit buffers are empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1" . When data is set to the transmit buffers, the bit is cleared to "0".

9.10 Error Flag

Three error flags are provided in the SCxCR register. The meaning of the flags is changed depending on the modes. The table below shows the meanings in each mode.

These flags are cleared to "0" after reading the SCxCR register.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART	Overrun error	Parity error	Framing error
I/O Interface (SCLK input)	Overrun error	Underrun error (When using double buffer or FIFO)	Fixed to 0
		Fixed to 0 (When a double buffer and FIFO unused)	
I/O Interface (SCLK output)	Undefined	Undefined	Fixed to 0

9.10.1 OERR Flag

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

In the I/O interface with SCLK output mode, the SCLK output stops upon setting the flag.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the overrun flag.

9.10.2 PERR Flag

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

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

In the I/O interface mode, <PERR> is set to "1" under the following conditions when a double buffer is enabled.

In the SCLK input mode, <PERR> is set to "1" when the SCLK is input after completing data output of the transmit shift register with no data in the transmit buffer.

In the SCLK output mode, <PERR> is set to "1" after completing output of all data and the SCLK output stops.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the under-run flag.

9.10.3 FERR Flag

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

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

9.11 Receive

9.11.1 Receive Counter

The receive counter is a 4-bit binary counter and is up-counted by SIOCLK. In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

9.11.2 Receive Control Unit

9.11.2.1 I/O interface mode

In the SCLK output mode with SCxCR <IOC> set to "0", the RXD pin is sampled on the rising edge of the shift clock outputted to the SCLK pin.

In the SCLK input mode with SCxCR <IOC> set to "1", the serial receive data RXD pin is sampled on the rising or falling edge of SCLK input signal depending on the SCxCR <SCLKS> setting.

9.11.2.2 UART Mode

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

9.11.3 Receive Operation

9.11.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is "0" cleared by reading the receive buffer. The receive buffer full flag does not have any value for the single buffer.

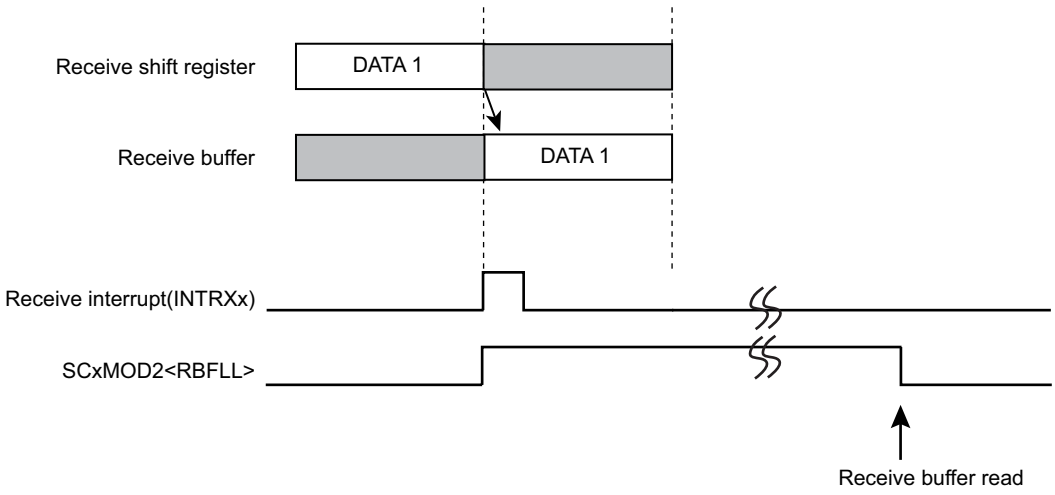


Figure 9-4 Receive Buffer Operation

9.11.3.2 Receive FIFO Operation

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

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

The following describes configurations and operations in the half duplex RX mode.

SCxMOD1[6:5] =01	: Transfer mode is set to half duplex mode
SCxFCNF[4:0] = 10111	: Automatically inhibits continuous reception after reaching the fill level. : The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxRFC[1:0] = 00	: The fill level of FIFO in which generated receive interrupt is set to 4-byte.
SCxRFC[7:6] = 11	: Clears receive FIFO and sets the condition of interrupt generation.

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

In this above condition, if the continuous reception after reaching the fill level is enabled, and it is possible to receive a data continuously with and reading the data in the FIFO.

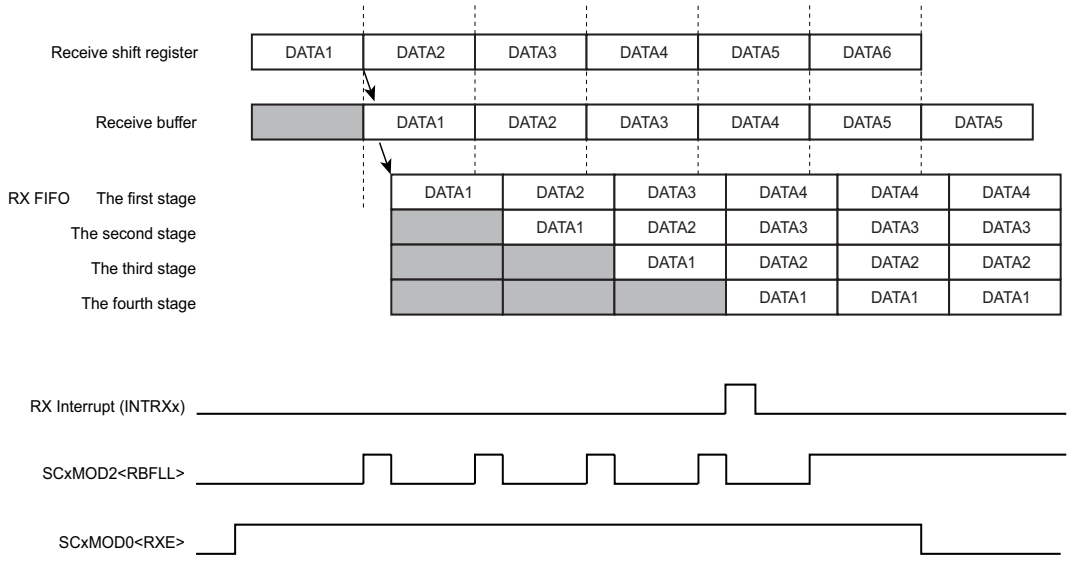


Figure 9-5 Receive FIFO Operation

9.11.3.3 I/O interface mode with SCLK output

In the I/O interface mode and SCLK output setting, SCLK output stops when all received data is stored in the receive buffer and FIFO. So, in this mode, the overrun error flag has no meaning.

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

(1) Case of single buffer

Stop SCLK output after receiving a data. In this mode, I/O interface can transfer each data with the transfer device by hand-shake.

When the data in a buffer is read, SCLK output is restarted.

(2) Case of double buffer

Stop SCLK output after receiving the data into a receive shift register and a receive buffer.

When the data is read, SCLK output is restarted.

(3) Case of FIFO

Stop SCLK output after receiving the data into a shift register, received buffer and FIFO.

When one byte data is read, the data in the received buffer is transferred into FIFO and the data in the receive shift register is transferred into received buffer and SCLK output is restarted.

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

9.11.3.4 Read Received Data

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

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. In the case of the next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

When the receive FIFO is available, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

9.11.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1." In this case, the interrupt INTRXx will be generated only when SCxCR <RB8> is set to "1."

9.11.3.6 Overrun Error

When FIFO is disabled, the overrun error is occurred and set overrun flag without completing data read before receiving the next data. When overrun error is occurred, a content of receive buffer and SCxCR<RB8> is not lost, but a content of receive shift register is lost.

When FIFO is enabled, overrun error is occurred and set overrun flag by no reading the data before moving the next data into received buffer when FIFO is full. In this case, the content of FIFO are not lost.

In the I/O interface mode with SCLK output setting, the clock output automatically stops, so this flag has no meaning.

Note: When the mode is changed from I/O interface SCLK output mode to the other mode, read SCxCR and clear overrun flag.

9.12 Transmission

9.12.1 Transmission Counter

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

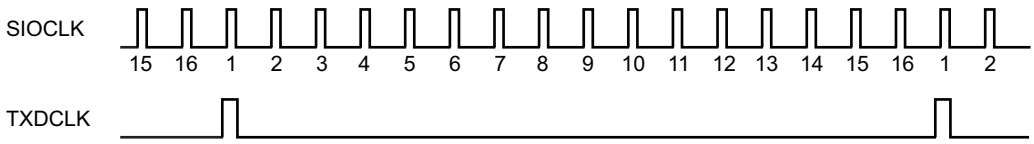


Figure 9-6 Generation of Transmission Clock

9.12.2 Transmission Control

9.12.2.1 I/O Interface Mode

In the SCLK output mode with SCxCR<IOC> set to "0", each bit of data in the transmit buffer is outputted to the TXD pin on the falling edge of the shift clock outputted from the SCLK pin.

In the SCLK input mode with SCxCR<IOC> set to "1", each bit of data in the transmit buffer is outputted to the TXD pin on the rising or falling edge of the SCLK input signal according to the SCxCR<SCLKS> setting.

9.12.2.2 UART Mode

When the transmit data is written in the transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock signal is also generated.

9.12.3 Transmit Operation

9.12.3.1 Operation of Transmission Buffer

If double buffering is disabled, the CPU writes data only to Transmit shift register and the transmit interrupt INTTXx is generated upon completion of data transmission.

If double buffering is enabled (including the case the transmit FIFO is enabled), data written to the transmit buffer is moved to the transmit shift register. The INTTXx interrupt is generated at the same time and the transmit buffer empty flag (SCxMOD2<TBEMP>) is set to "1". This flag indicates that the next transmit data can be written. When the next data is written to the transmit buffer, the <TBEMP> flag is cleared to "0".

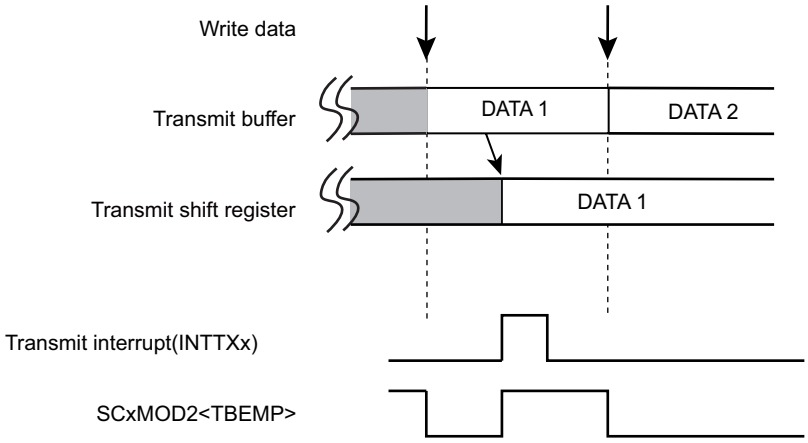


Figure 9-7 Operation of Transmission Buffer (Double-buffer is enabled)

9.12.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

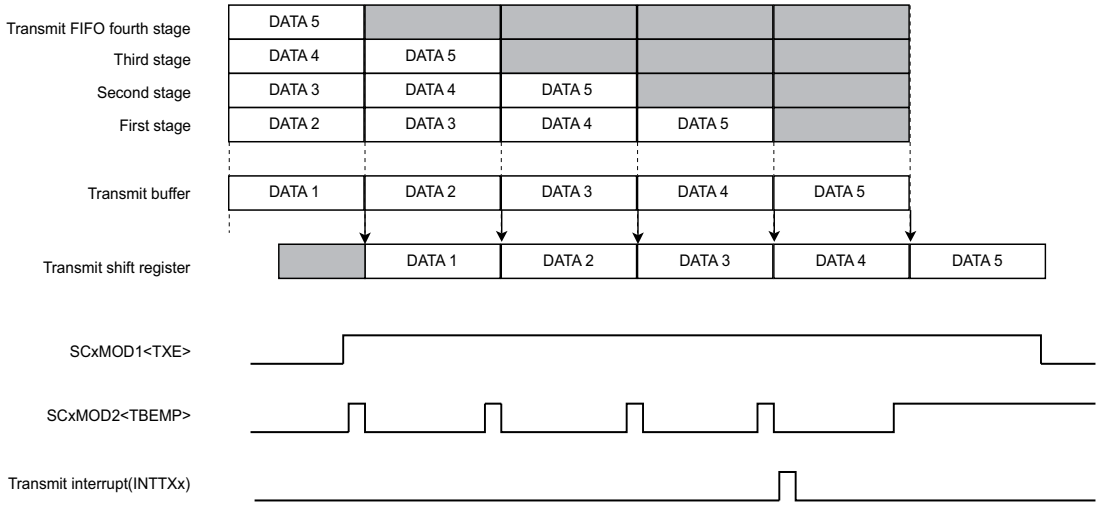
Note: **To use TX FIFO buffer, TX FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG> = "1").**

Settings and operations to transmit 4-byte data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1[6:5] =10	: Transfer mode is set to half duplex.
SCxFCNF[4:0] = 11011	: Transmission is automatically disabled if FIFO becomes empty. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.
SCxTFC[1:0] = 00	: Sets the interrupt generation fill level to "0".
SCxTFC[7:6] = 11	: Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF[0] = 1	: Enable FIFO.

After above settings are configured, data transmission can be initiated by writing 5 bytes of data to the transmit buffer and FIFO, and setting the SCxMOD1<TXE> bit to "1". When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the transmission sequence is terminated.

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should lasts by writing transmit data.



9.12.3.3 I/O interface Mode/Transmission by SCLK Output

If SCLK is set to generate clock the I/O interface mode, the SCLK output automatically stops when all data transmission is completed and underrun error will not occur.

The timing of suspension and resume of SCLK output is different depending on the buffer and FIFO usage.

(1) Single Buffer

The SCLK output stops each time one frame of data is transferred. Handshaking for each data with the other side of communication can be enabled. The SCLK output resumes when the next data is written in the buffer.

(2) Double Buffer

The SCLK output stops upon completion of data transmission of the transmit shift register and the transmit buffer. The SCLK output resumes when the next data is written in the buffer.

(3) FIFO

The transmission of all data stored in the transmit shift register, transmit buffer and FIFO is completed, the SCLK output stops. The next data is written, SCLK output resumes.

If SCxFCNF<RXTXCNT> is configured, SCxMOD0<TXE> bit is cleared at the same time as SCLK stop and the transmission stops.

9.12.3.4 Under-run error

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

In the I/O interface mode with SCLK output setting, the clock output automatically stops, so this flag has no meaning.

Note: **Before switching the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the underrun flag.**

9.13 Handshake function

The function of the handshake is to enable frame-by-frame data transmission by using the CTS (Clear to send) pin and to prevent overrun errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{CTS}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTS}}$ pin returns to the "Low" level. However in this case, the INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note 1: If the $\overline{\text{CTS}}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.(Point "a" in Figure 9-9)

Note 2: **Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "L".**(Point "b" in Figure 9-9)

Although no $\overline{\text{RTS}}$ pin is provided, a handshake control function can easily implemented by assigning one bit of the port for the $\overline{\text{RTS}}$ function. By setting the port to "High" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

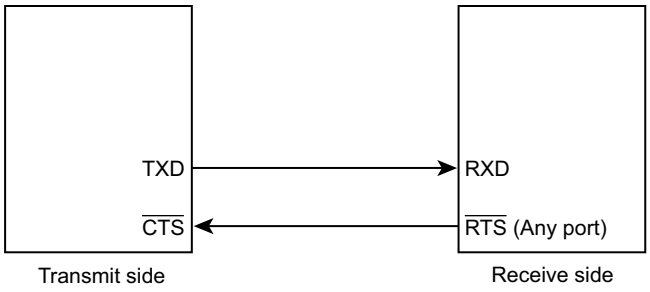


Figure 9-8 Handshake Function

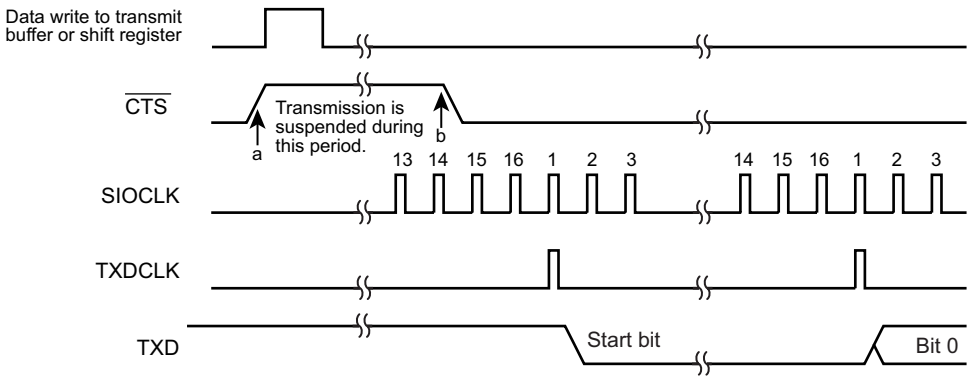


Figure 9-9 $\overline{\text{CTS}}$ Signal timing

9.14 Interrupt/Error Generation Timing

9.14.1 RX Interrupts

Figure 9-10 shows the data flow of receive operation and the route of read.

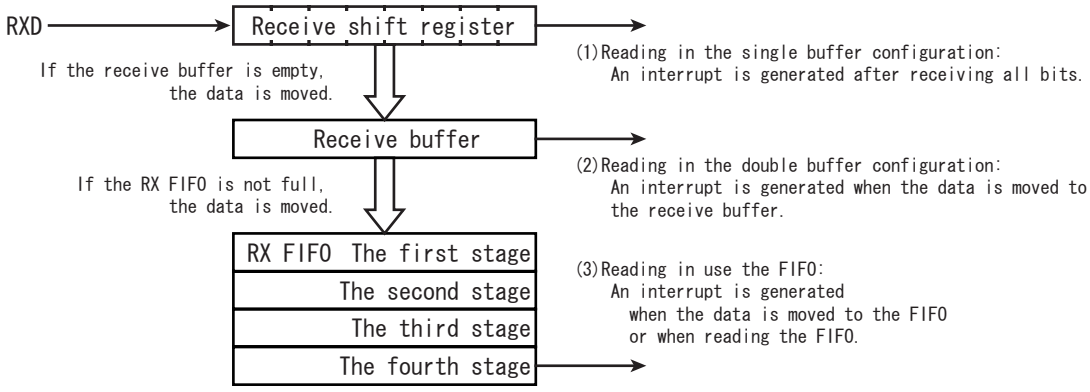


Figure 9-10 Receive Buffer/FIFO Configuration Diagram

9.14.1.1 Single Buffer / Double Buffer

RX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configurations	UART modes	IO interface modes
Single Buffer	–	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	Around the center of the first stop bit	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.) On data transfer from the shift register to the buffer by reading buffer.

Note: Interrupts are not generated when an overrun error is occurred.

9.14.1.2 FIFO

In use of FIFO, receive interrupt is generated on the condition that the following either operation and SCxRFC<RFIS > setting are established.

- Reception completion of all bits of one frame.
- Reading FIFO

Interrupt conditions are decided by the SCxRFC<RFIS> settings as described in Table 9-11.

Table 9-11 Receive Interrupt conditions in use of FIFO

SCxRFC<RFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is greater than or equal to "the fill level of FIFO intrruption generation."

9.14.2 TX interrupts

Figure 9-11 shows the data flow of transmit operation and the route of read.

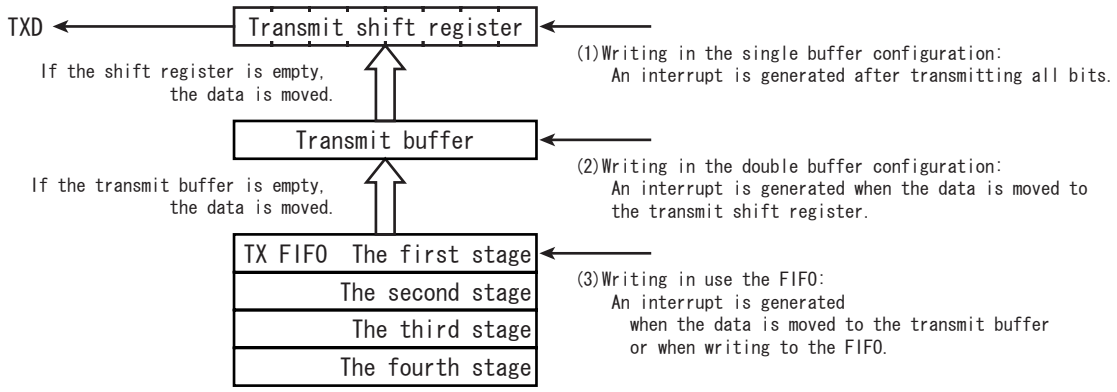


Figure 9-11 Transmit Buffer/FIFO Configuration Diagram

9.14.2.1 Single Buffer / Double Buffer

TX interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register.	

Note: If double buffer is enabled, a interrupt is also generated when the data is moved from the buffer to the shift register by writing to the buffer.

9.14.2.2 FIFO

In use of FIFO, transmit interrupt is generated on the condition that the following either operation and SCxTFC<TFIS> setting are established.

- Transmittion completion of all bits of one frame.
- Writing FIFO

Interrupt conditions are decided by the SCxTFC<TFIS> settings as described in Table 9-12.

Table 9-12 Transmit Interrupt conditions in use of FIFO

SCxTFC<TFIS>	Interrupt conditions
"0"	"The fill level of FIFO" is equal to "the fill level of FIFO interruption generation."
"1"	"The fill level of FIFO" is smaller than or equal to "the fill level of FIFO interruption generation."

9.14.3 Error Generation

9.14.3.1 UART Mode

modes	9 bits	7 bits 8 bits 7 bits+ Parity 8 bits + Parity
Framing Error Overrun Error	Around the center of stop bit	
Parity Error	–	Around the center of parity bit

9.14.3.2 IO Interface Mode

Overrun Error	Immediately after the rising / falling edge of the last SCLK (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Underrun Error	Immediately after the rising or falling edge of the next SCLK. (Rising or falling is determined according to SCxCR<SCLKS> setting.)

Note: Over-run error and Under-run error have no meaning in SCLK output mode.

9.15 Software Reset

Software reset is generated by writing SCxMOD2<SWRST[1:0]> as "10" followed by "01".

As a result, SCxMOD0<RXE>, SCxMOD1<TXE>, SCxMOD2<TBEMP><RBFLL><TXRUN>, SCxCR <OERR><PERR><FERR> are initialized. And the receive circuit, the transmit circuit and the FIFO become initial state. Other states are held.

9.16 Operation in Each Mode

9.16.1 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.

9.16.1.1 Transmitting Data

(1) SCLK Output Mode

- If the transmit double buffer is disabled (SCxMOD2<WBUF> = "0")

  Data is output from the TXD pin and the clock is output from the SCLK pin each time the CPU writes data to the transmit buffer. When all data is output, an interrupt (INTTXx) is generated.

- If the transmit double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer while data transmission is halted or when data transmission from the transmit buffer (shift register) is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  When data is moved from the transmit buffer to the transmit shift register, if the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the SCLK output stops.

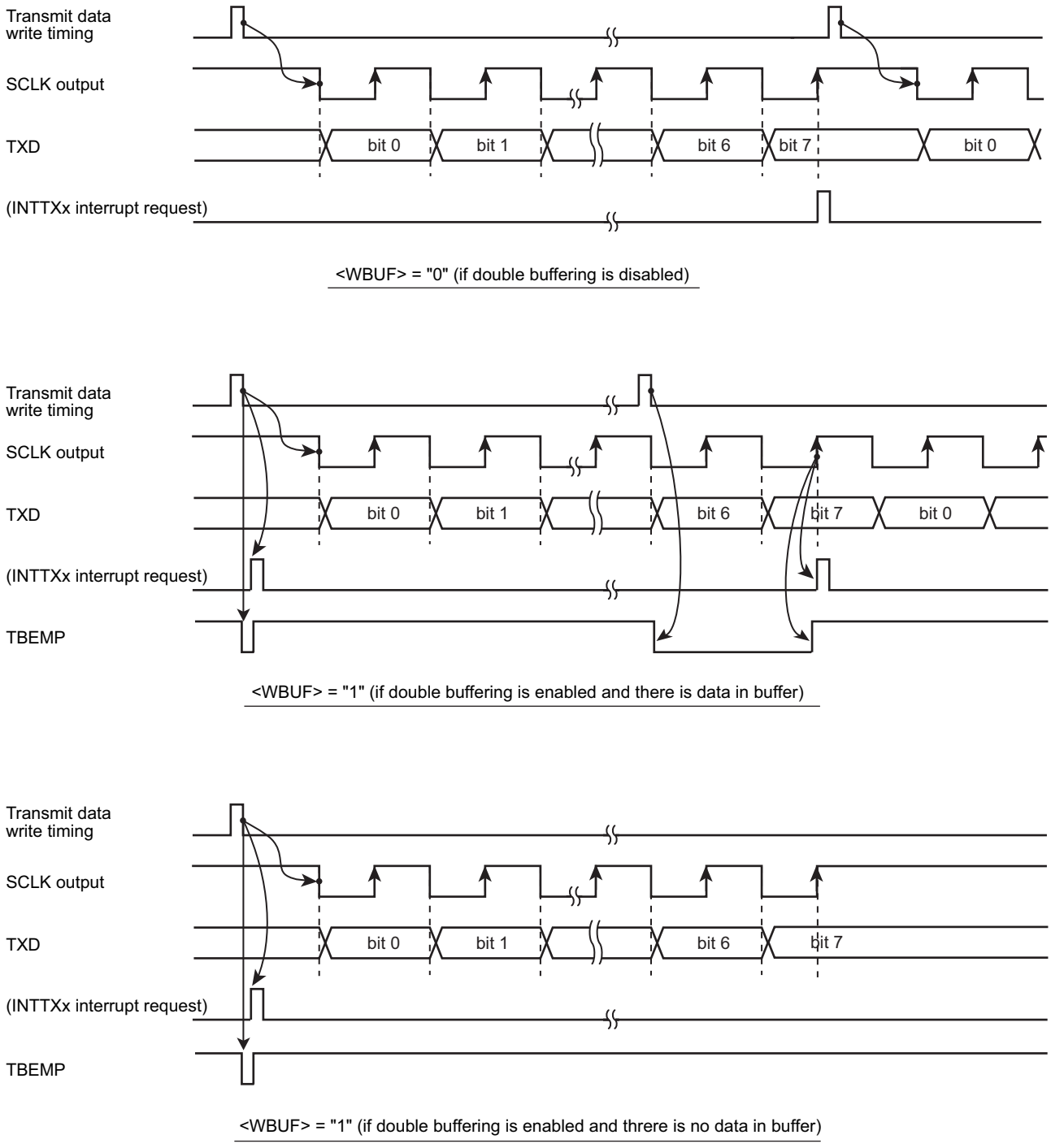


Figure 9-12 Transmit Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the SCLK is input in the condition where data is written in the transmit buffer, 8-bit data is outputted from the TXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing point "A" as shown in Figure 9-13.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the SCLK input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, the transmit buffer empty flag SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the SCLK input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (0xFF) is sent.

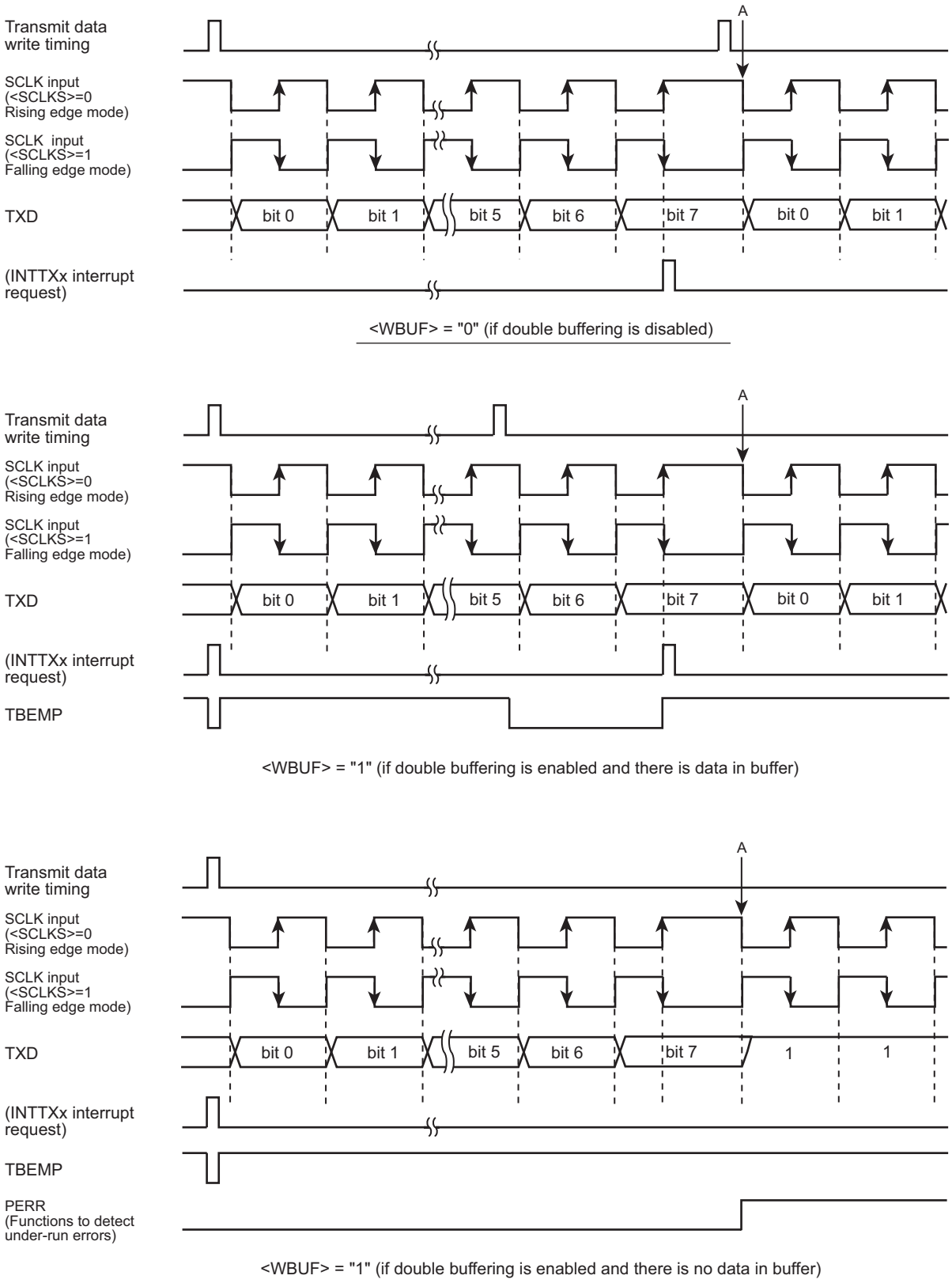


Figure 9-13 Transmit Operation in the I/O Interface Mode (SCLK Input Mode)

9.16.1.2 Receive

(1) SCLK Output Mode

The SCLK output can be started by setting the receive enable bit SCxMOD0<RXE> to "1".

- If double buffer is disabled (SCxMOD2<WBUF> = "0")

  A clock pulse is outputted from the SCLK pin and the next data is stored into the shift register each time the CPU reads received data. When all the 8 bits are received, the INTRXx interrupt is generated.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data stored in the shift register is moved to the receive buffer and the receive buffer can receive the next frame. A data is moved from the shift register to the receive buffer, the receive buffer full flag SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  While data is in the receive buffer, if the data cannot be read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the SCLK output stops. In this state, reading data from the receive buffer allows data in the shift register to move to the receive buffer and thus the INTRXx interrupt is generated and data reception resumes.

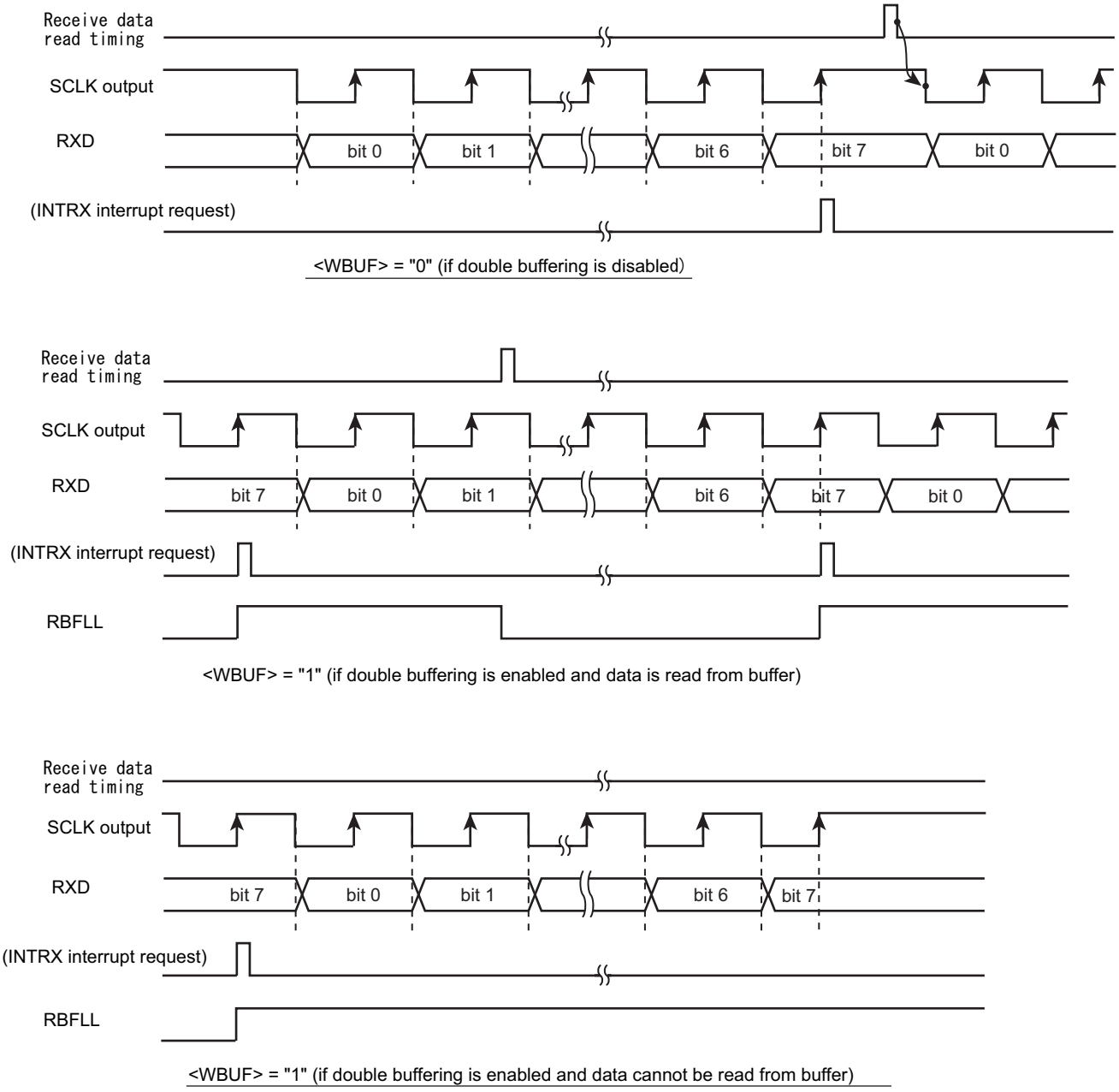


Figure 9-14 Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

In the SCLK input mode, receiving double buffering is always enabled, the received frame can be moved to the receive buffer from the shift register, and the receive buffer can receive the next frame successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.

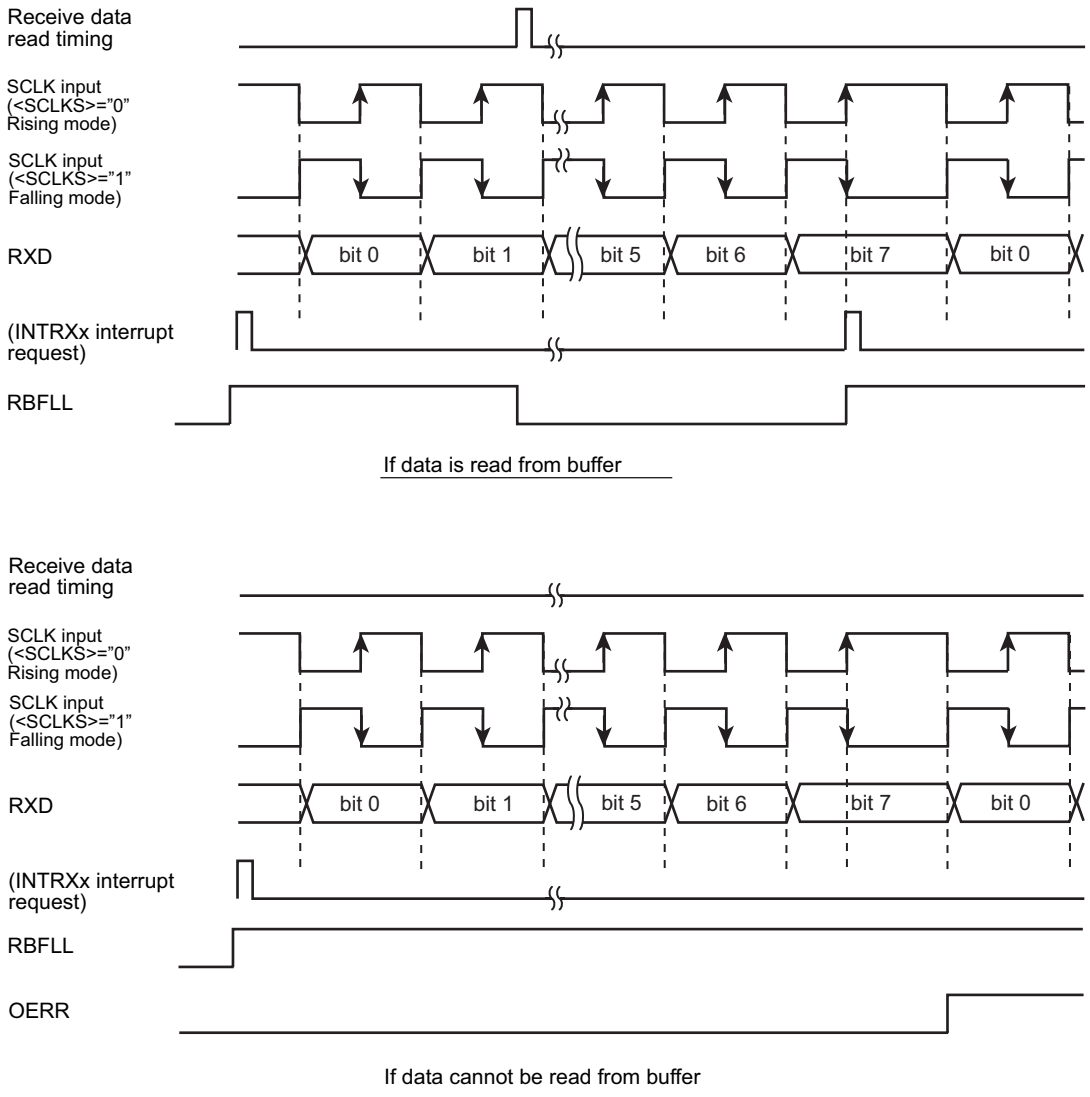


Figure 9-15 Receive Operation in the I/O Interface Mode (SCLK Input Mode)

9.16.1.3 Transmit and Receive (Full-duplex)

(1) SCLK Output Mode

- If SCxMOD2<WBUF> is set to "0" and the double buffers are disabled

  SCLK is outputted when the CPU writes data to the transmit buffer.

  Subsequently, 8 bits of data are shifted into receive shift register and the INTRXx receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are outputted from the TXD pin, the INTTXx transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops.

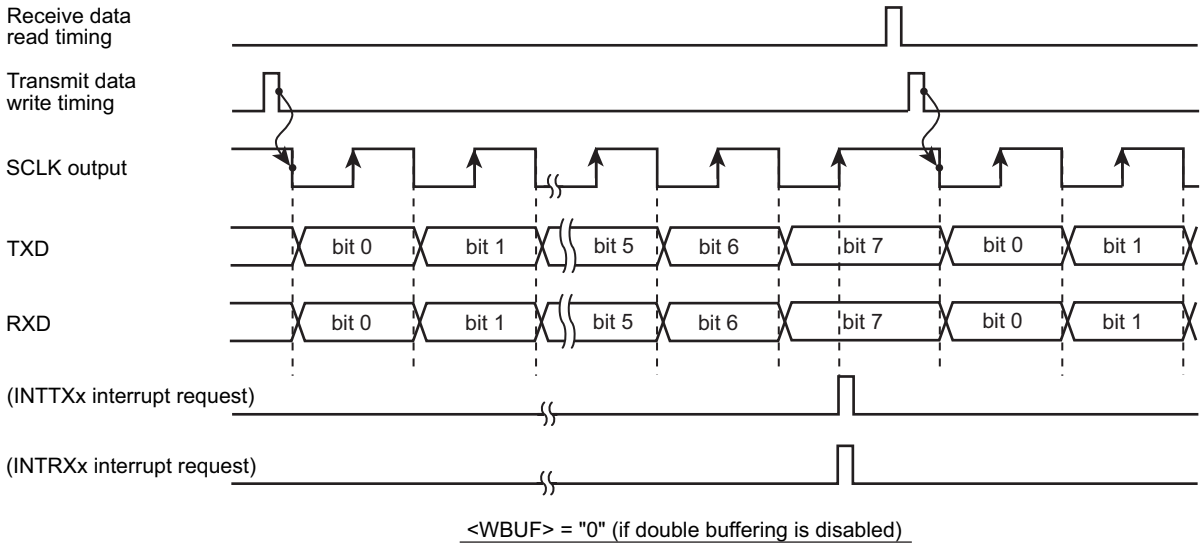
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If SCxMOD2<WBUF> is set to "1" and the double buffers are enabled

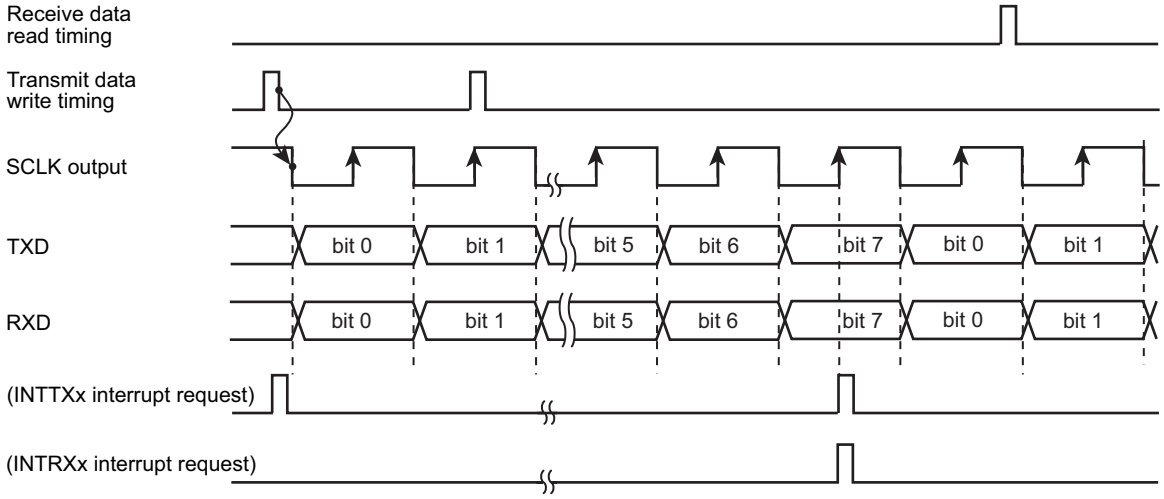
  SCLK is outputted when the CPU writes data to the transmit buffer.

  8 bits of data are shifted into the receive shift register, moved to the receive buffer, and the INTRXx interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is outputted from the TXD pin. When all data bits are sent out, the INTTXx interrupt is generated and the next data is moved from the transmit buffer to the transmit shift register.

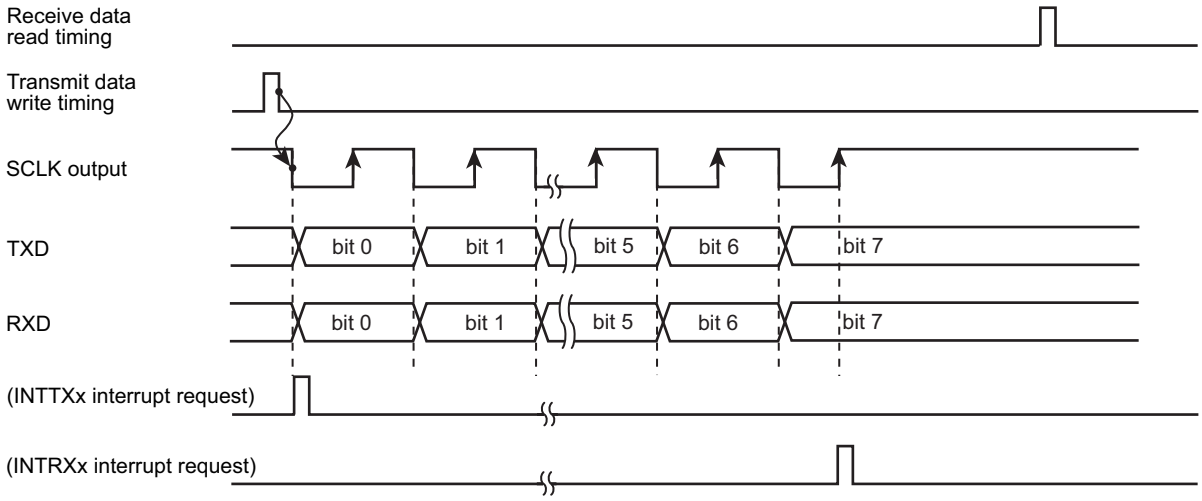
  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = 1) or when the receive buffer is full (SCxMOD2<RBFULL> = 1), the SCLK output is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the next round of data transmission and reception is started.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled)



<WBUF> = "1" (if double buffering is enabled)

Figure 9-16 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) SCLK Input Mode

- If SCxMOD2<WBUF> is set to "0" and the transmit double buffer is disabled

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  8-bit data written in the transmit buffer is outputted from the TXD pin and 8 bit of data is shifted into the receive buffer when the SCLK input becomes active.The INTTXx interrupt is generated upon completion of data transmission. The INTRXx interrupt is generated when the data is moved from receive shift register to receive buffer after completion of data reception.

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 9-17). Data must be read before completing reception of the next frame data.

- If SCxMOD2<WBUF> is set to "1" and the double buffer is enabled.

  The interrupt INTTXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx interrupt is generated.

  Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Figure 9-17). Data must be read before completing reception of the next frame data.

  Upon the SCLK input for the next frame, transmission from transmit shift register (in which data has been moved from transmit buffer) is started while receive data is shifted into receive shift register simultaneously.

  If data in receive buffer has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to transmit buffer when SCLK for the next frame is input, an under-run error occurs.

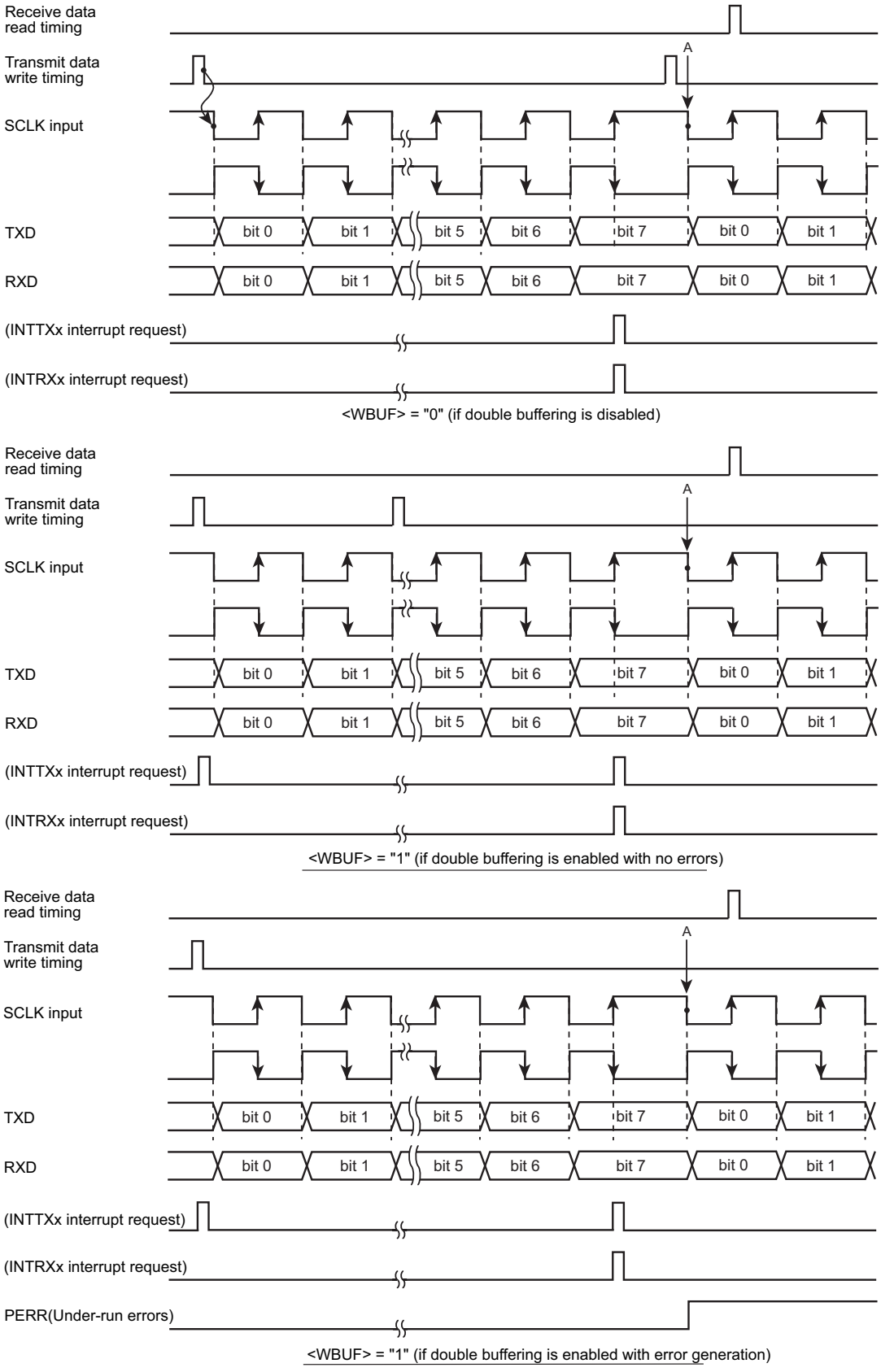


Figure 9-17 Transmit/Receive Operation in the I/O Interface Mode (SCLK Input Mode)

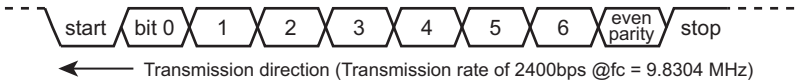
9.16.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode can be selected by setting the mode control register (SCxMOD<SM[1:0]>) to "01".

In this mode, parity bits can be added to the transmit data stream; the control register (SCxCR<PE>) controls the parity enable/disable setting.

When <PE> is set to "1" (enable), either even or odd parity may be selected using the SCxCR<EVEN> bit. The length of the stop bit can be specified using SCxMOD2<SBLEN>.

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



Clocking conditions	system clock :	High-speed (fc)
	High-speed clock gear:	×1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

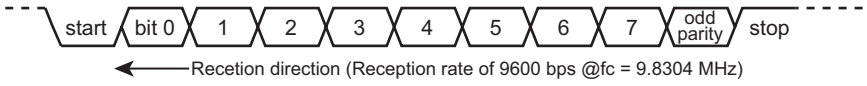
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x : don't care - : no change

9.16.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SCxCR<PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SCxCR<EVEN>.

The control register settings for receiving data in the following format are as follows:



Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	×1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	Set 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x : don't care - : no change

9.16.4 Mode 3 (9-bit UART mode)

The 9-bit UART mode can be selected by setting SCxMOD0<SM[1:0]> to "11." In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to SCxMOD0<TB8> for transmitting data. The data is stored in SCxCR<RB8>.

When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SCxBUF.

The stop bit length can be specified using SCxMOD2<SBLEN>.

9.16.4.1 Wake up function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SCxMOD0<WU> to "1."

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: The TXD pin of the slave controller must be set to the open drain output mode using the PxOD register.

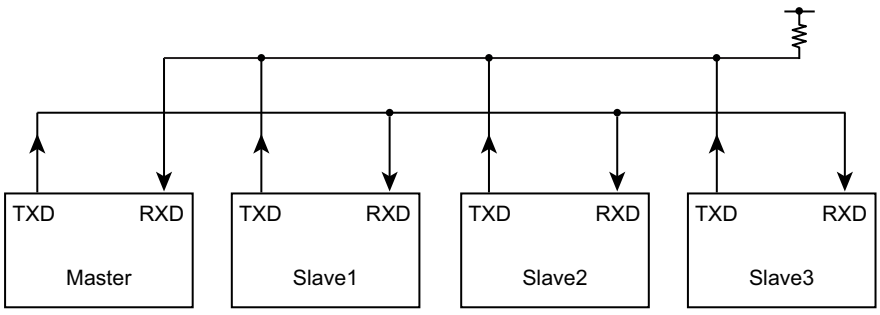


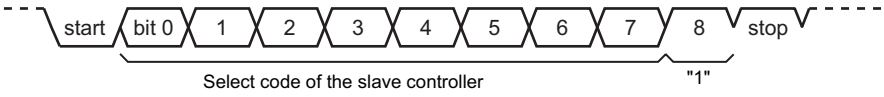
Figure 9-18 Serial Links to Use Wake-up Function

9.16.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

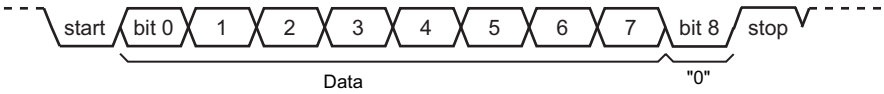
2. Set SCxMOD0<WU> to "1" for the slave controllers to make them ready to receive data.

3. The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1".



4. Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0".



6. The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRXx) is generated.Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

10. Serial Bus Interface (I2C/SIO)

The TMPM376FDDFG/FDFG contains one Serial Bus Interface (I2C/SIO) channels, in which the following two operating modes are included:

- I2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I2C bus mode, the I2C/SIO is connected to external devices via SCL and SDA.

In the clock-synchronous 8-bit SIO mode, the I2C/SIO is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the I2C/SIO in each operating mode.

Table 10-1 Port settings for using serial bus interface

channel	Operating mode	pin	Port Function Register	Port Output Control Register	Port Input Control Register	Port Open Drain Output Control Register
SBI	I2C bus mode	SCL :PN1 SDA :PN0	PNFR1[1:0] = 11	PNCR[1:0] = 11	PNIE[1:0] = 11	PNOD[1:0] = 11
	SIO mode	SCK :PN2 SI :PN1 SO :PN0	PNFR1[2:0] = 111	PNCR[2:0] = 101(SCK0 output) PNCR[2:0] = 001(SCK0 input)	PNIE[2:0] = 010(SCK0 output) PNIE[2:0] = 110(SCK0 input)	PNOD[2:0] = xxx

Note: x: Don't care

10.1 Configuration

The configuration is shown in Figure 10-1.

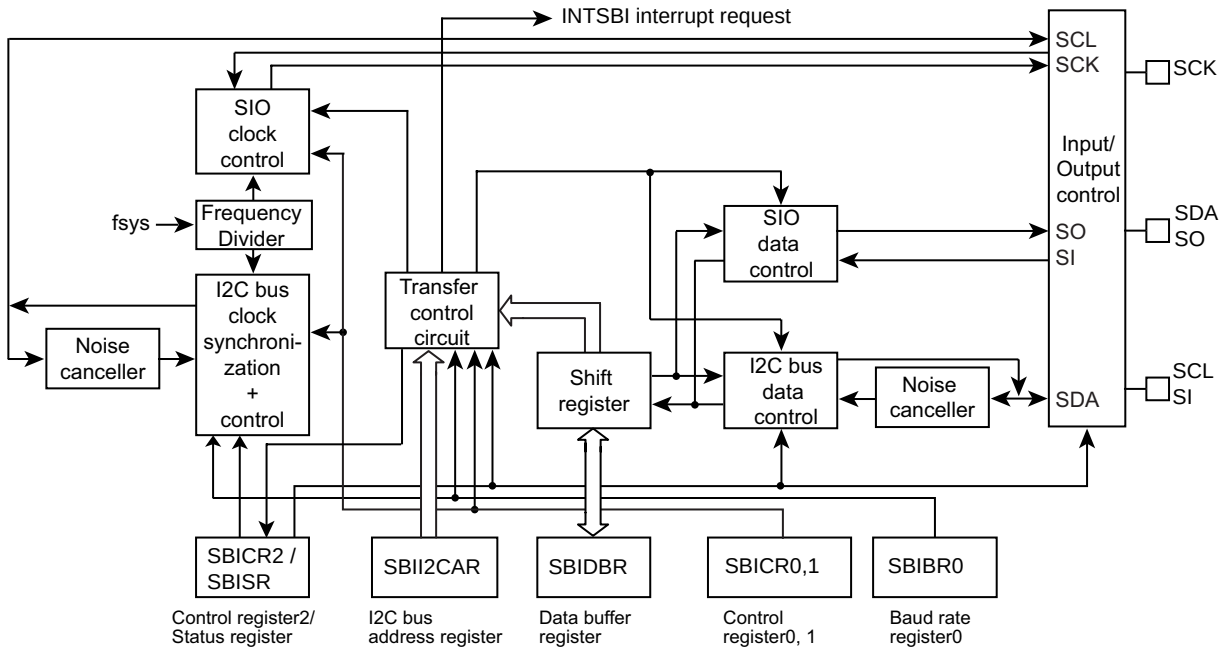


Figure 10-1 (I2C/SIO) Block Interface

10.2 Register

The following registers control the serial bus interface and provide its status information for monitoring.

The register below performs different functions depending on the mode. For details, refer to "10.4 Control Registers in the I2C Bus Mode" and "10.7 Control register of SIO mode".

10.2.1 Registers for each channel

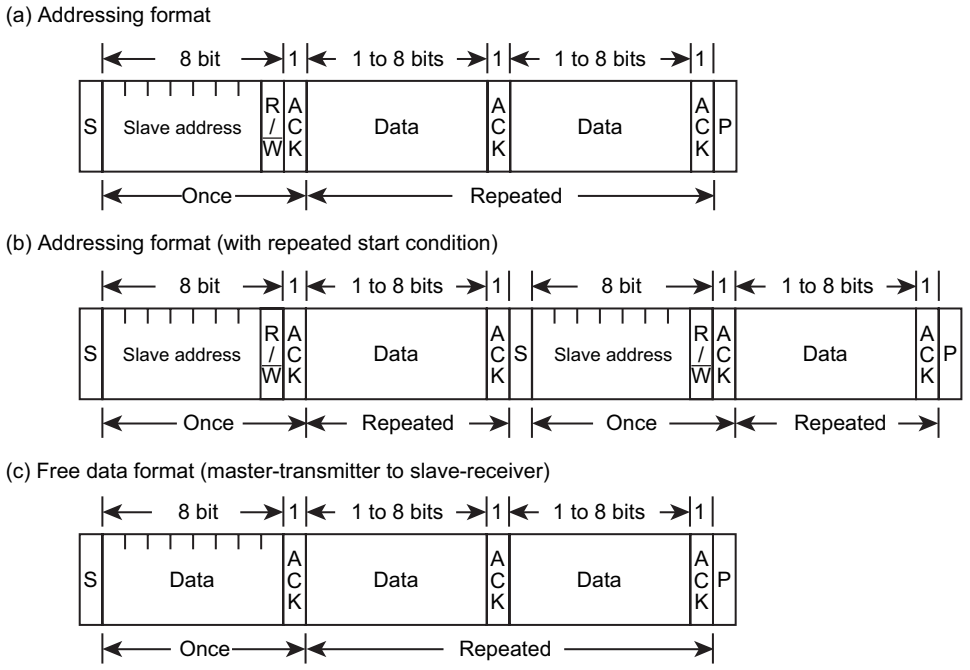
The tables below show the registers and register addresses for each channel.

Base Address = 0x4002_0000

Register name		Address(Base+)
Control register 0	SBICR0	0x0000
Control register 1	SBICR1	0x0004
Data buffer register	SBIDBR	0x0008
I2C bus address register	SBII2CAR	0x000C
Control register 2	SBICR2 (writing)	0x0010
Status register	SBISR (reading)	
Baud rate register 0	SBIBR0	0x0014

10.3 I2C Bus Mode Data Format

Figure 10-2 shows the data formats used in the I2C bus mode.



Note) S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 10-2 I2C Bus Mode Data Formats

10.4 Control Registers in the I2C Bus Mode

The following registers control the serial bus interface in the I2C bus mode and provide its status information for monitoring.

10.4.1 SBICR0(Control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBICR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it's been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

Note: To use the serial bus interface, enable this bit first.

10.4.2 SBICR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
After reset	0	0	0	0	1	0	0	1(Note3)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-5	BC[2:0]	R/W	Select the number of bits per transfer (Note 1)
4	ACK	R/W	Master mode 0: Acknowledgement clock pulse is not generated. 1: Acknowledgement clock pulse is generated. Slave mode 0: Acknowledgement clock pulse is not counted. 1: Acknowledgement clock pulse is counted.
3	-	R	Read as 1.
2-1	SCK[2:1]	R/W	Select internal SCL output clock frequency (Note 2).
0	SCK[0]	W	
	SWRMON	R	On reading <SWRMON>: Software reset status monitor 0:Software reset operation is in progress. 1:Software reset operation is not in progress.

<BC>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

000	n = 5	769kHz	System Clock: fsys (= 80MHz)
001	n = 6	588kHz	Clock gear : fc/1
010	n = 7	400 kHz	Frequency = $\frac{fsys}{2^n + 72}$ [Hz]
011	n = 8	244 kHz	
100	n = 9	137 kHz	
101	n = 10	73 kHz	
110	n = 11	38 kHz	
111		reserved	

Note 1: **Clear <BC[2:0]> to "000" before switching the operation mode to the SIO mode.**

Note 2: **For details on the SCL line clock frequency, refer to "10.5.1 Serial Clock".**

Note 3: **After a reset, the <SCK[0]/SWRMON> bit is read as "1". However, if the SIO mode is selected at the SBICR2 register, the initial value of the <SCK[0]> bit is "0".**

Note 4: **The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.**

Note 5: When <BC[2:0]>="001" and <ACK>="0" in master mode, SCL line may be fixed to "L" by falling edge of SCL line after generation of STOP condition and the other devices can not use the bus. In the case of bus which is connected with several master devices, the bumber of bits per transfer should be set equal or more than 2 before generation of STOP condition.

10.4.3 SBICR2(Control register 2)

This register serves as SBISR register by reading it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBI interrupt request 0: – 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. When writing, set <SBIM[1:0]> to "10"; I2Cbus mode.

Note: **Make sure that modes are not changed during a communication session.Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.**

10.4.4 SBISR (Status Register)

This register serves as SBICR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBI interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: – 1:Detected
2	AAS	R	Slave address match detection 0: – 1: Detected (This bit is set when the general call is detected as well.)
1	ADO	R	General call detection 0: – 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

10.4.5 SBIBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

10.4.6 SBIDBR (Serial bus interface data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R (Receive)/ W (Transmit)	Receive data / Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIBDR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

10.4.7 SBII2CAR (I2Cbus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: **Please set the bit 0 <ALS> of I2C bus address register SBII2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.**

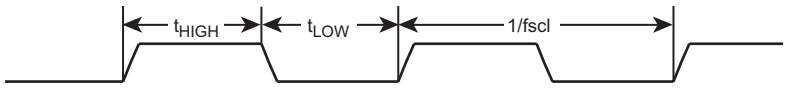
Note 2: **Do not set SBII2CAR to "0x00" in slave mode. (If SBII2CAR is set to "0x00", it's recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)**

10.5 Control in the I2C Bus Mode

10.5.1 Serial Clock

10.5.1.1 Clock source

SBICR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$t_{LOW} = 2^{n-1}/fsys + 58/fsys$
$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$
$fscl = 1/(t_{LOW} + t_{HIGH})$
$= \frac{fsys}{2^n + 72}$

SBICR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 10-3 Clock source

Note: **The maximum speeds in the standard and high-speed modes are specified to 100kHz and 400kHz respectively following the communications standards. Notice that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.**

10.5.1.2 Clock Synchronization

The I2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "Low" level overrides other masters producing the "High" level on their clock lines. This must be detected and responded by the masters producing the "High" level.

Clock synchronization assures correct data transfer on a bus that has two or more master.

For example, the clock synchronization procedure for a bus with two masters is shown below.

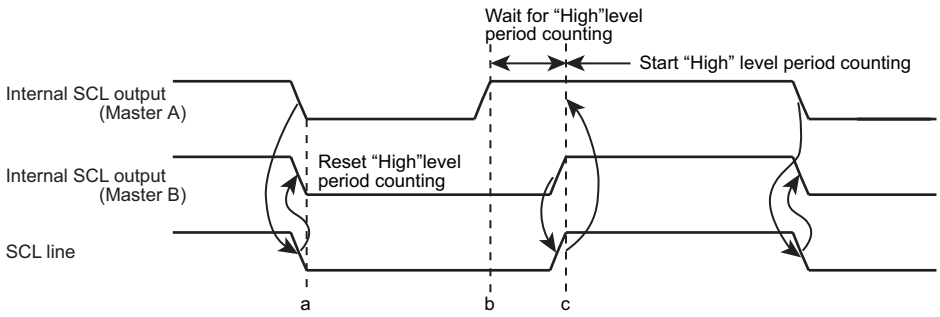


Figure 10-4 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the "Low" level, bringing the SCL bus line to the "Low" level. Master B detects this transition, resets its "High" level period counter, and pulls its internal SCL output level to the "Low" level.

Master A completes counting of its "Low" level period at the point b, and brings its internal SCL output to the "High" level. However, Master B still keeps the SCL bus line at the "Low" level, and Master A stops counting of its "High" level period counting.After Master A detects that Master B brings its internal SCL output to the "High" level and brings the SCL bus line to the "High" level at the point c, it starts counting of its "High" level period.

After that Master finishes counting the "High" level period, the Master pulls the SCL pin to "Low" and the SCL bus line becomes "Low".

This way, the clock on the bus is determined by the master with the shortest "High" level period and the master with the longest "Low" level period among those connected to the bus.

10.5.2 Setting the Acknowledgement Mode

Setting SBICR1<ACK> to "1" selects the acknowledge mode.When operating as a master, the SBI adds one clock for acknowledgment signal. In slave mode, the clock for acknowledgement signals is counted. In transmitter mode, the SBI releases the SDAx pin during clock cycle to receive acknowledgement signals from the receiver. In receiver mode, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals. Also in slave mode, if a general-call address is received, the SBI pulls the SDAx pin to the "Low" level during the clock cycle and generates acknowledgement signals.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is counted.

10.5.3 Setting the Number of Bits per Transfer

SBICR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

10.5.4 Slave Addressing and Address Recognition Mode

Setting "0" to SBII2CAR<ALS> and a slave address in SBII2CAR<SA[6:0]> sets addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

10.5.5 Operating mode

The setting of SBICR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, ensure that the serial bus interface pins are at "High" level before setting <SBIM[1:0]> to "10". Also, ensure that the bus is free before switching the operating mode to the port mode.

10.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBICR2<TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

At the slave mode:

- when data is transmitted in the addressing format.
- when the received slave address matches the value specified at SBII2CAR.
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros.

If the value of the direction bit (R/$\overline{W}$) is "1", <TRX> is set to "1" by the hardware. If the bit is "0", <TRX> is set to "0".

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0", <TRX> changes to "1". If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

If SBI is used in free data format, <TRX> is not changed by the hardware.

10.5.7 Configuring the SBI as a Master or a Slave

Setting SBICR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

10.5.8 Generating Start and Stop Conditions

When SBISR<BB> is "0", writing "1" to SBICR2<MST, TRX, BB, PIN> causes the SBI to start a sequence for generating the start condition and to output the slave address and the direction bit prospectively written in the data buffer register. <ACK> must be set to "1" in advance.

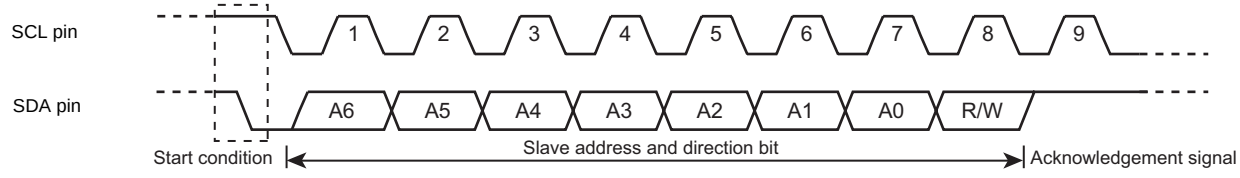


Figure 10-5 Generating the Start Condition and a Slave Address

When <BB> is "1", writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

If SCL bus line is pulled "Low" by other devices when the stop condition is generated, the stop condition is generated after the SCL line is released.

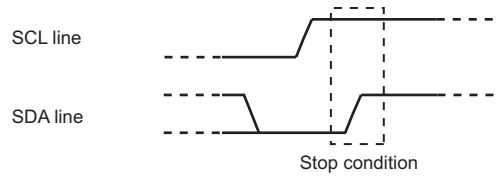


Figure 10-6 Generating the Stop Condition

SBISR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and cleared to "0" when the stop condition is detected (the bus is free).

10.5.9 Interrupt Service Request and Release

In master mode, a serial bus interface request (INTSBI) is generated when the transfer of the number of clock cycles set by <BC> and <ACK> is completed.

In slave mode, INTSBI is generated under the following conditions.

- After output of the acknowledge signal which is generated when the received slave address matches the slave address set to SBII2CAR<SA[6:0]>.
- After the acknowledge signal is generated when a general-call address is received.
- When the slave address matches or a data transfer is completed after receiving a general-call address.

In the address recognition mode (<ALS> = "0"), INTSBI is generated when the received slave address matches the values specified at SBII2CAR or when a general-call (eight bits data following the start condition is all "0") is received.

When an interrupt request (INTSBI) is generated, SBICR2<PIN> is cleared to "0". While <PIN> is cleared to "0", the SBI pulls the SCL line to the "Low" level.

<PIN> is set to "1" when data is written to or read from SBIDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1". When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear this bit to "0".

Note: When arbitration is lost in master mode, <PIN> is not cleared to "0" if the slave address does not match (INTSBI is generated).

10.5.10 Arbitration Lost Detection Monitor

The I2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines.The I2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below.

Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the "Low" level and Master B outputs the "High" level.

Then Master A pulls the SDA bus line to the "Low" level because the line has the wired-AND connection. When the SCL line goes high at the point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid.

This condition of Master B is called "Arbitration Lost". Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

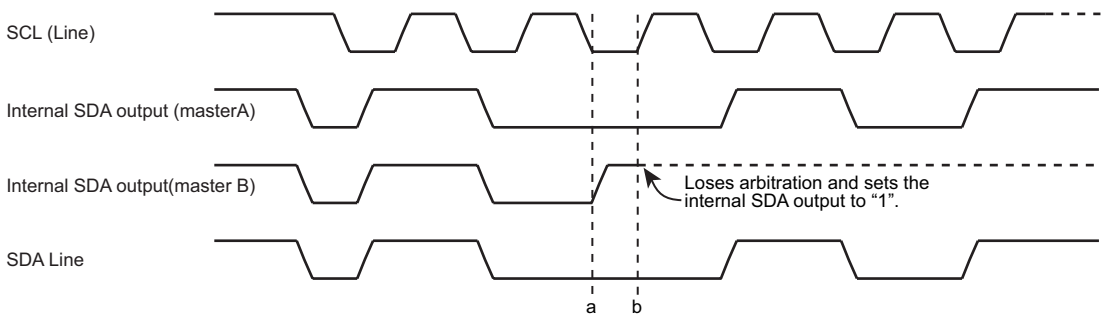


Figure 10-7 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs and SBISR<AL> is set to "1".

When <AL> is set to "1", SBISR<MST, TRX> are cleared to "0", causing the SBI to operate as a slave receiver.Therefore, the serial bus interface circuit stops the clock output during data transfer after <AL> is set to "1".

<AL> is cleared to "0" when data is written to or read from SBIDBR or data is written to SBICR2.

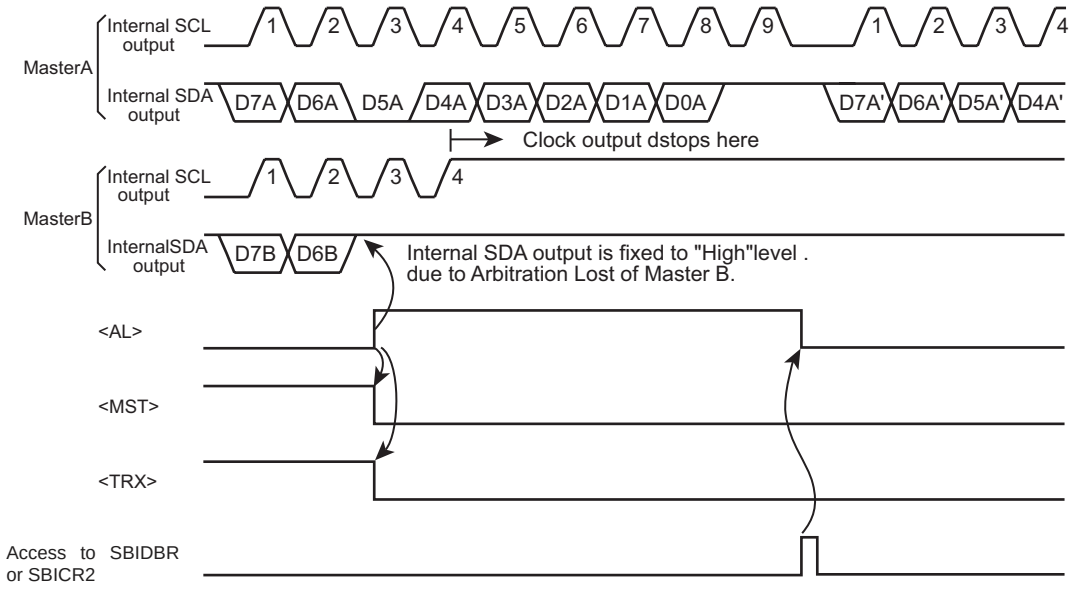


Figure 10-8 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

10.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBII2CAR<ALS>="0"), SBISR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBII2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIDBR.

10.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBISR<ADO> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros.

<ADO> is cleared to "0" when the start or stop condition is detected on the bus.

10.5.13 Last Received Bit Monitor

SBISR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBISR<LRB> immediately after generation of the INTSBI interrupt request causes ACK signal to be read.

10.5.14 Data Buffer Register (SBIDBR)

Reading or writing SBIDBR initiates reading received data or writing transmitted data.

When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

10.5.15 Baud Rate Register (SBIBR0)

The SBIBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

10.5.16 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset. Writing "10" followed by "01" to SBICR2<SWRST[1:0]> generates a reset signal that initializes the serial bus interface circuit. When writing, set <SBIM[1:0]> to "10"; I2Cbus mode. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to "0".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

10.6 Data Transfer Procedure in the I2C Bus ModeI2C

10.6.1 Device Initialization

First, program SBICR1<ACK, SCK[2:0]>. Writing "000" to SBICR1<BC[2:0]> at the time.

Next, program SBII2CAR by specifying a slave address at <SA[6:0]> and an address recognition mode at <ALS>. (<ALS> must be cleared to "0" when using the addressing format).

To configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "0" to SBICR2<MST, TRX, BB>, "1" to <PIN>, "10" to <SBIM[1:0]> and "0" to the bit 1 and 0.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBICR1	←	0	0	0	X	0	X	X	X	Specifies ACK and SCL clock.
SBII2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBICR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X; Don't care

10.6.2 Generating the Start Condition and a Slave Address

10.6.2.1 Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBICR1<ACK> to select the acknowledgment mode. Write to SBIDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBICR2<MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the master mode, the SBI holds the SCL line at the "Low" level while <PIN> is = "0".<TRX> changes its value according to the transmitted direction bit at generation of the INTSBI interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

```
                        7  6  5  4  3  2  1  0
Reg.          ←  SBISR
Reg.          ←  Reg. e 0x20
if Reg.       ≠  0x00                          Ensures that the bus is free.
Then

SBICR1        ←  X  X  X  1  0  X  X  X        Selects the acknowledgement mode.
SBIDBR        ←  X  X  X  X  X  X  X  X        Specifies the desired slave address and direction.
SBICR2        ←  1  1  1  1  1  0  0  0        Generates the start condition.
Example of INTSBI0 interrupt routine
    Clears the interrupt request.
    Processing
    End of interrupt
```

10.6.2.2 Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line.

If the received address matches its slave address specified at SBII2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "Low" level during the ninth clock and outputs an acknowledgment signal.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0". In the slave mode, the SBI holds the SCL line at the "Low" level while <PIN> is "0".

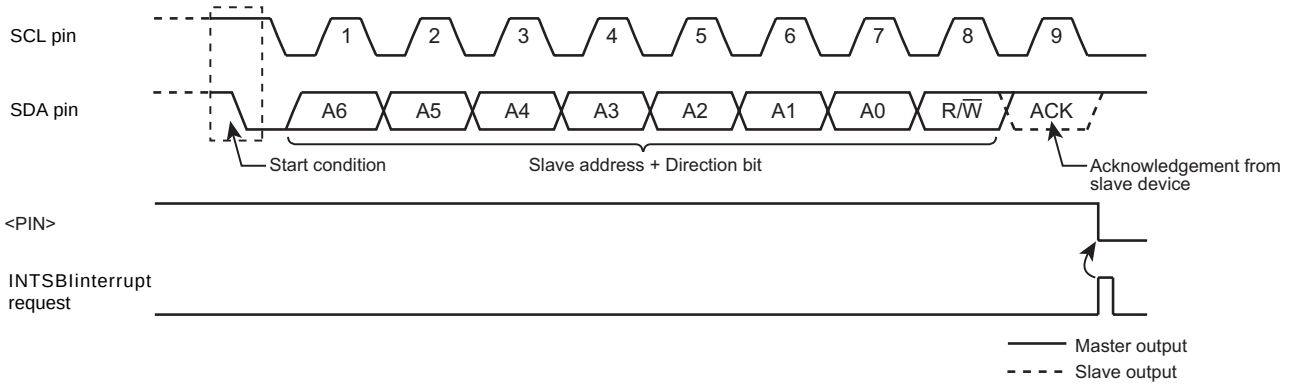


Figure 10-9 Generation of the Start Condition and a Slave Address

10.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBI interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

10.6.3.1 Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(1) Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1", that means the receiver requires no further data.

The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0", that means the receiver requires further data.If the next data to be transmitted has eight bits, the data is written into SBIDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIDBR.Writing the data makes <PIN> to "1", causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word.

After the transfer is completed, the INTSBI interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBI interrupt

if MST = 0

Then go to the slave-mode processing.

if TRX = 0

Then go to the receiver-mode processing.

if LRB = 0

Then go to processing for generating the stop condition.

SBICR1	←	X	X	X	X	0	X	X	X	Specifies the number of bits to be transmitted and specify whether ACK is required.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

End of interrupt processing.

Note: **X; Don't care**

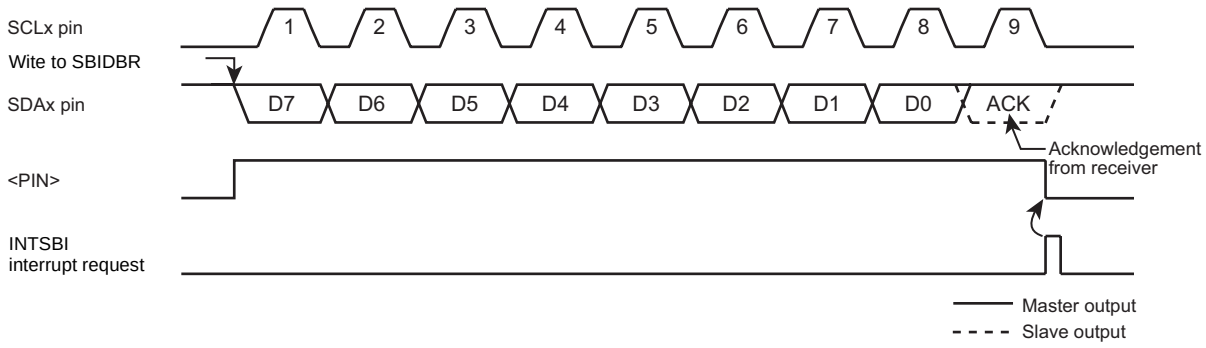


Figure 10-10 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(2) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIDBR.

If the data has different length, <BC[2:0]> and <ACK> are programmed and the received data is read from SBIDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)On reading the data, <PIN> is set to "1", and the serial clock is output to the SCL pin to transfer the next data word.In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SDA pin.

After that, the INTSBI interrupt request is generated, and <PIN> is cleared to "0", pulling the SCL pin to the "Low" level.Each time the received data is read from SBIDBR, one-word transfer clock and an acknowledgement signal are output.

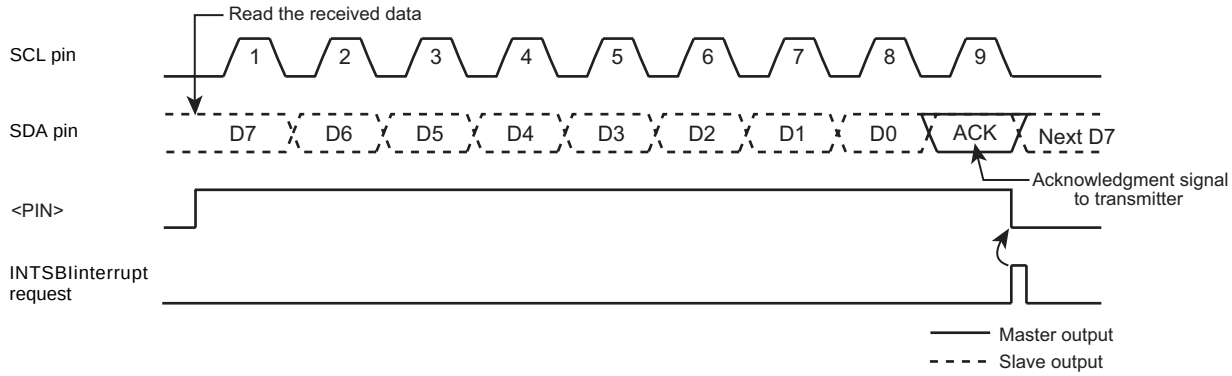


Figure 10-11 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be cleared to "0" immediately before reading the data word second to last.

This disables generation of an acknowledgment clock for the last data word.

When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC[2:0]> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer.

At this time, the master receiver holds the SDA bus line at the "High" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

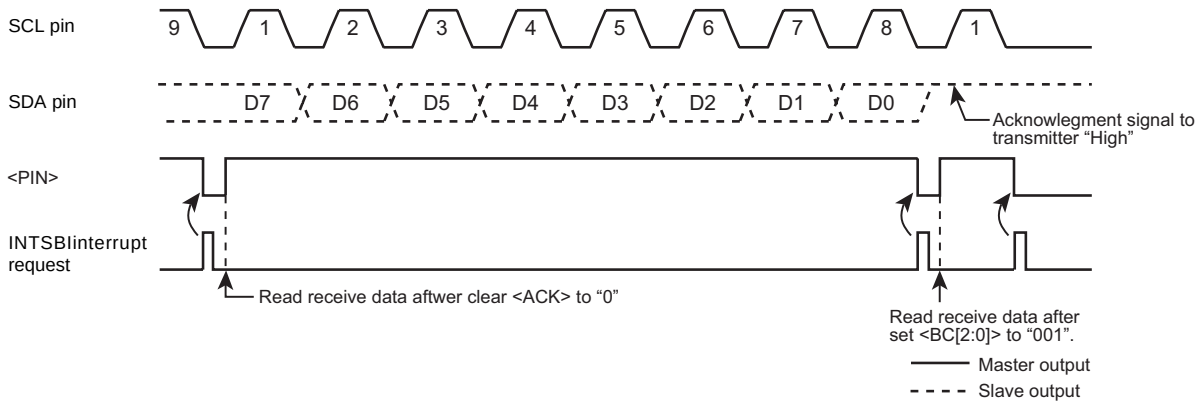


Figure 10-12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBI interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBICR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIDBR								Reads dummy data.
End of interrupt										

INTSBI interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIDBR								Reads the first to (N-2)th data words.
End of interrupt										

INTSBI interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBICR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIDBR								Reads the (N-1)th data word.
End of interrupt										

INTSBI interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIDBR								Reads the Nth data word.
End of interrupt										

INTSBI interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.
End of interrupt	

Note: X; Don't care

10.6.3.2 Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBI interrupt request on four occasions:

1) when the SBI has received any slave address from the master.

2) when the SBI has received a general-call address.

3) when the received slave address matches its address.

4) when a data transfer has been completed in response to a general-call.

Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode.

Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBI interrupt request is generated, <PIN> is cleared to "0", and the SCL pin is pulled to the "Low" level.

When data is written to or read from SBIDBR or when <PIN> is set to "1", the SCLx pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBISR<AL>, <TRX>, <AAS> and <ADO> are tested to determine the processing required.

"Table 10-2 Processing in Slave Mode"shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode.

INTSBI interrupt

if TRX = 0

Then go to other processing.

if AL = 0

Then go to other processing.

if AAS = 0

Then go to other processing.

SBICR1	←	X	X	X	1	0	X	X	X	Sets the number of bits to be transmitted.
SBIDBR	←	X	X	X	X	X	X	X	X	Sets the transmit data.

Note: X; Don't care

Table 10-2 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<ADO>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC[2:0]> and write the transmit data into SBIDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC[2:0].> and read the received data from SBIDBR.

10.6.4 Generating the Stop Condition

When SBISR<BB> is "1", writing "1" to SBICR2<MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus.

Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released.

After that, the SDA pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBICR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

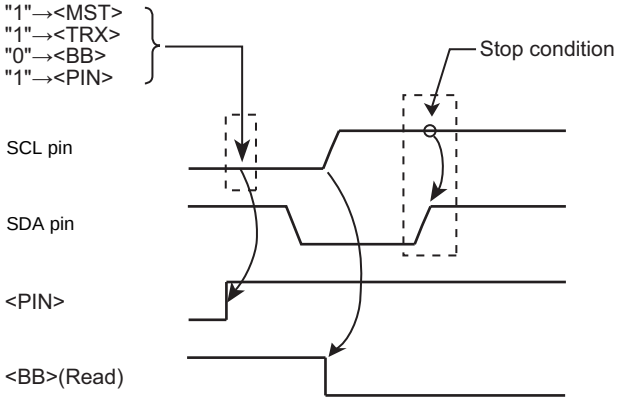


Figure 10-13 Generating the Stop Condition

10.6.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device.The procedure of generating a restart in the master mode is described below.

First, write SBICR2<MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDAx pin is held at the "High" level and the SCLx pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy.

Then, test SBISR<BB> and wait until it becomes "0" to ensure that the SCLx pin is released.

Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCLx bus line to the "Low" level.

Once the bus is determined to be free by following the above procedures, follow the procedures described in "10.6.2 Generating the Start Condition and a Slave Address"to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

Note 1: **Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)**

Note 2: **When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a restart becomes "1", the rising edge of the SCL line is not detected even <LBR>= "1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.**

		7	6	5	4	3	2	1	0	
SBICR2	←	0	0	0	1	1	0	0	0	Releases the bus.
if SBISR<BB> ≠ 0										Checks that the SCL pin is released.
Then										
if SBISR<LRB> ≠ 1										Checks that no other device is pulling the SCL pin to the "Low".
Then										
4.7 μs Wait										
SBICR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgment mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Sets the desired slave address and direction.
SBICR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Note: X; Don't care

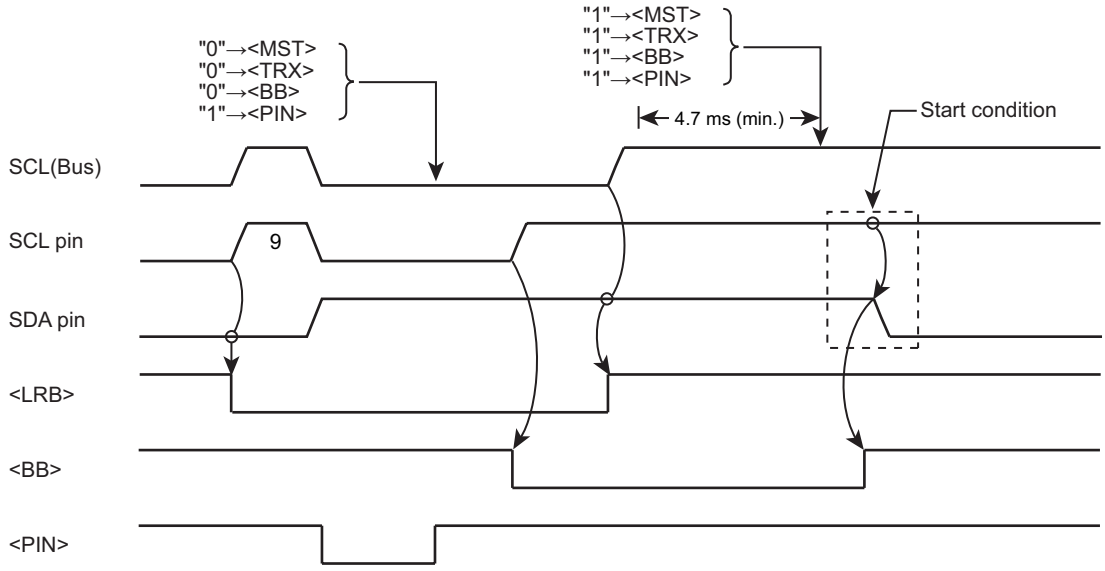


Figure 10-14 Timing Chart of Generating a Restart

10.7 Control register of SIO mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

10.7.1 SBICR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBICR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

10.7.2 SBICR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1)

000	n = 3	5 MHz	
001	n = 4	2.5 MHz	
010	n = 5	1.25 Hz	
011	n = 6	625 kHz	
100	n = 7	313 kHz	
101	n = 8	156 kHz	
110	n = 9	78 kHz	
111	–	External clock	

System clock: fsys (= 80MHz)
Clock gear: fc/1
Frequency = $\frac{fsys/2}{2^n}$ [Hz]

Note 1: **After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBICR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBICR2 register and the SBISR register are the same.**

Note 2: **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

10.7.3 SBIDBR (Data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: **The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.**

Note 2: **Since SBIDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.**

10.7.4 SBICR2(Control register 2)

This register serves as SBISR register by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

Note 2: **Make sure that modes are not changed during a communication session.**

10.7.5 SBISR (Status Register)

This register serves as SBICR2 by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-4	-	R	Read as 1.(Note)
3	SIOF	R	Serial transfer status monitor. 0: Completed 1: In progress
2	SEF	R	Shift operation status monitor 0: Completed. 1: In progress
1-0	-	R	Read as 1. (Note)

Note: **In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.**

10.7.6 SBIBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

10.8 Control in SIO mode

10.8.1 Serial Clock

10.8.1.1 Clock source

Internal or external clocks can be selected by programming SBICR1<SCK[2:0]>.

(1) Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCKx pin.

At the beginning of a transfer, the SCKx pin output becomes the "High" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

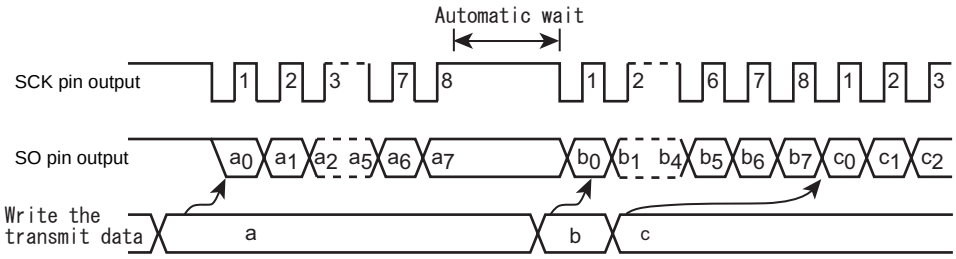
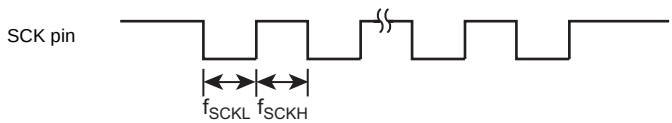


Figure 10-15 Automatic Wait

(2) External clock (<SCK[2:0]> = "111")

The SBI uses an external clock supplied from the outside to the SCKx pin as a serial clock.

For proper shift operations, the serial clock at the "High" and "Low" levels must have the pulse widths as shown below.



f_{SCKL}, f_{SCKH} > 4/fsys

Figure 10-16 Maximum Transfer Frequency of External Clock Input

10.8.1.2 Shift Edge

Leading-edge shift is used in transmission. Trailing-edge shift is used in reception.

- Leading-edge shift

  Data is shifted at the leading edge of the serial clock (or the falling edge of the SCKx pin input/output).

- Trailing-edge shift

  Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCKx pin input/output).

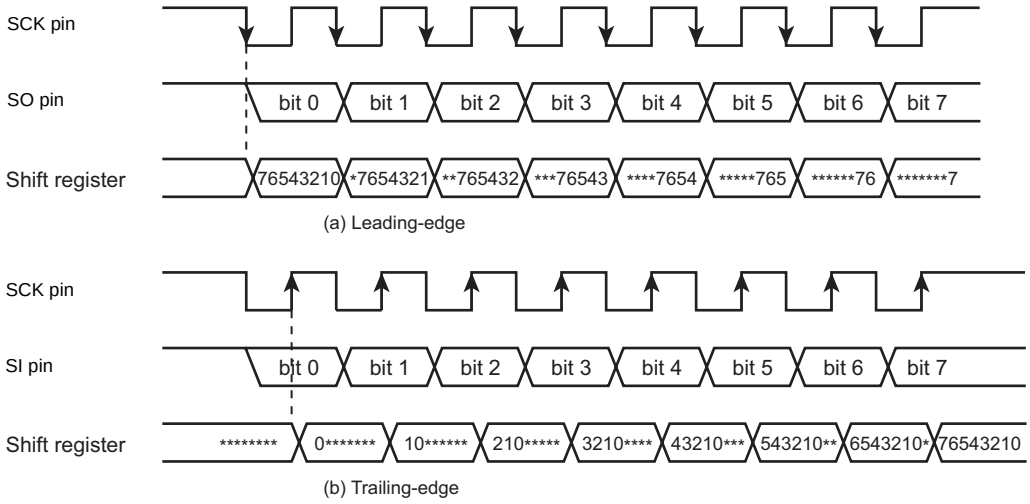


Figure 10-17 Shift Edge

10.8.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBICR1<SIOM[1:0]>.

10.8.2.1 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIDBR.

After writing the transmit data, writing "1" to SBICR1<SIOS> starts the transmission. The transmit data is moved from SBIDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIDBR becomes empty, and the INTSBI (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIDBR is loaded with the next transmit data.

In the external clock mode, SBIDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBISR<SIOF> to "1" to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBISR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBICR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBI interrupt

SBDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

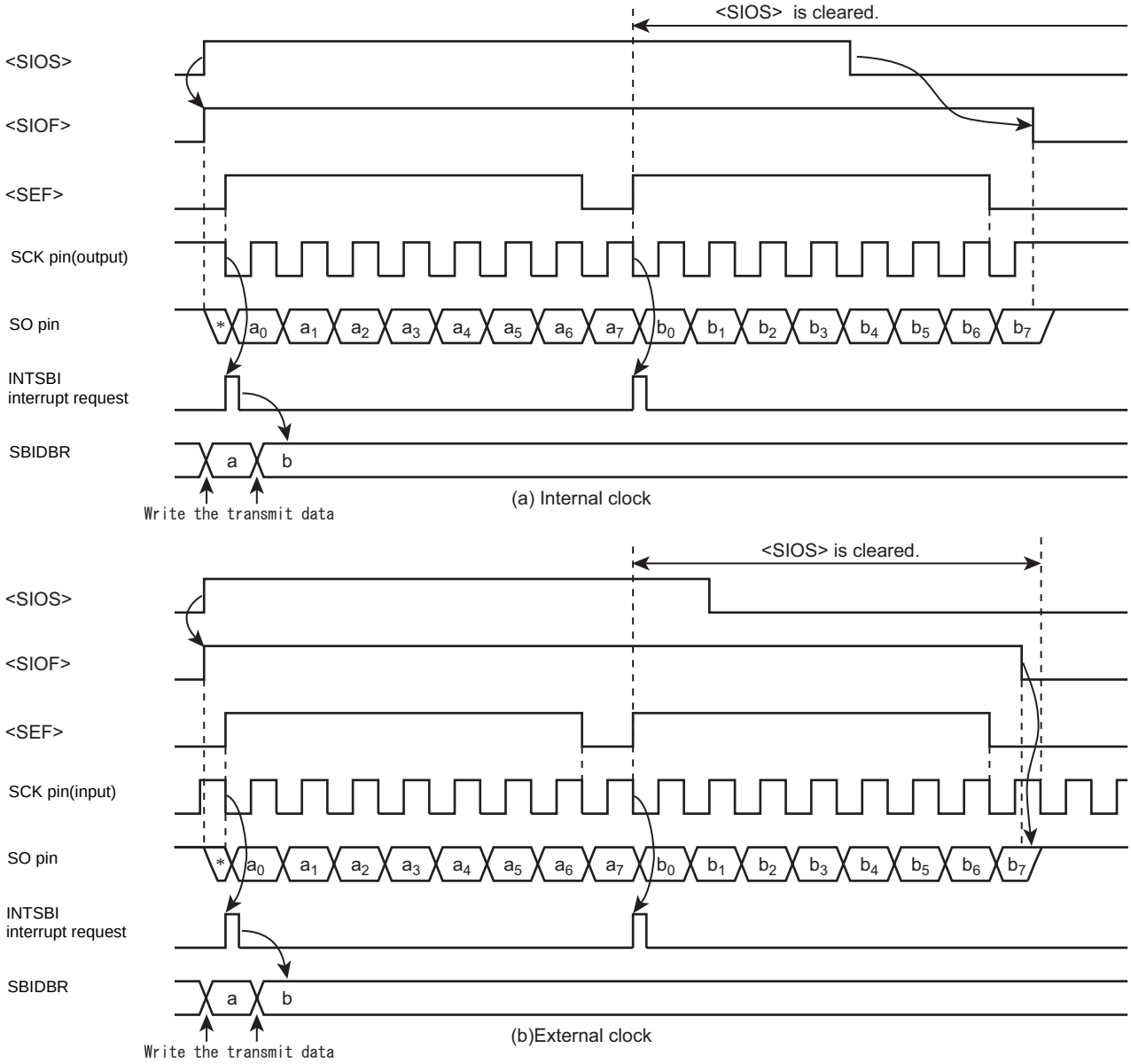


Figure 10-18 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

7 6 5 4 3 2 1 0

if SBISR<SIOF> ≠ 0 — Recognizes the completion of the transmission.

Then

if SCK ≠ 1 — Recognizes "1" is set to the SCK pin by monitoring the port.

Then

SBICR1 ← 0 0 0 0 0 1 1 1 — Completes the transmission by setting <SIOS> = 0.

10.8.2.2 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBICR1<SIOS> enables reception.Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIDBR and the INTSBI (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIDBR. The program checks SBISR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: **The contents of SBIDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBICR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBI interrupt

Reg.	←	SBIDBR	Reads the received data.

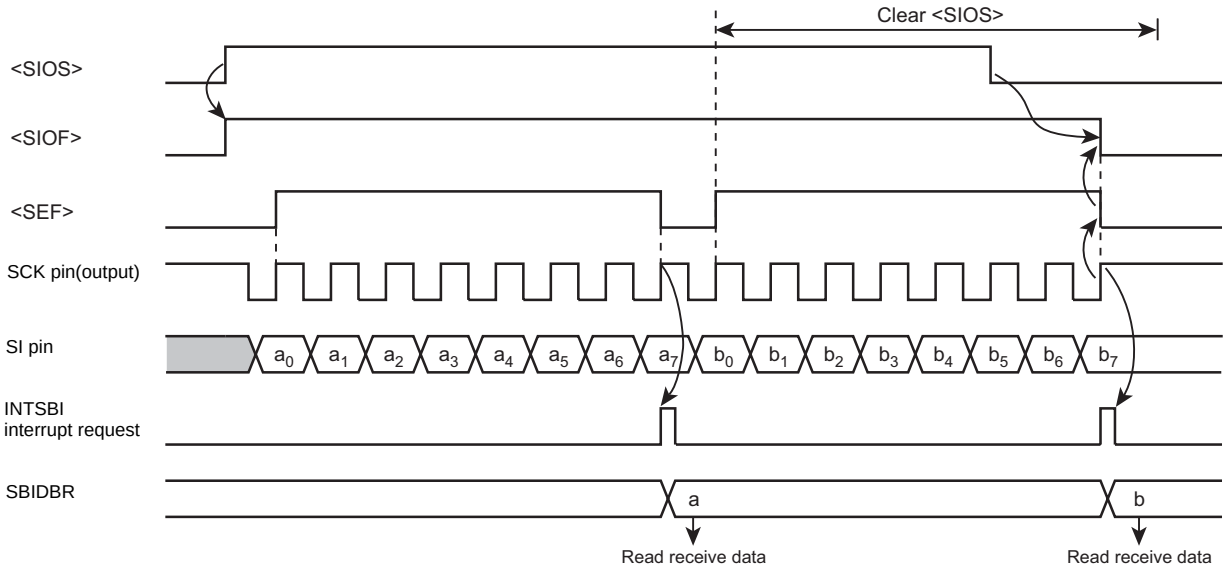


Figure 10-19 Receive Mode (Example: Internal Clock)

10.8.2.3 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIDBR and setting SBICR1<SIOS> to "1" enables transmission and reception.The transmit data is output through the SOx pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIDBR and the INTSBI interrupt request is generated.The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started.The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBICR1<SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIDBR. The program checks SBISR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception.If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: **The contents of SBIDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.**

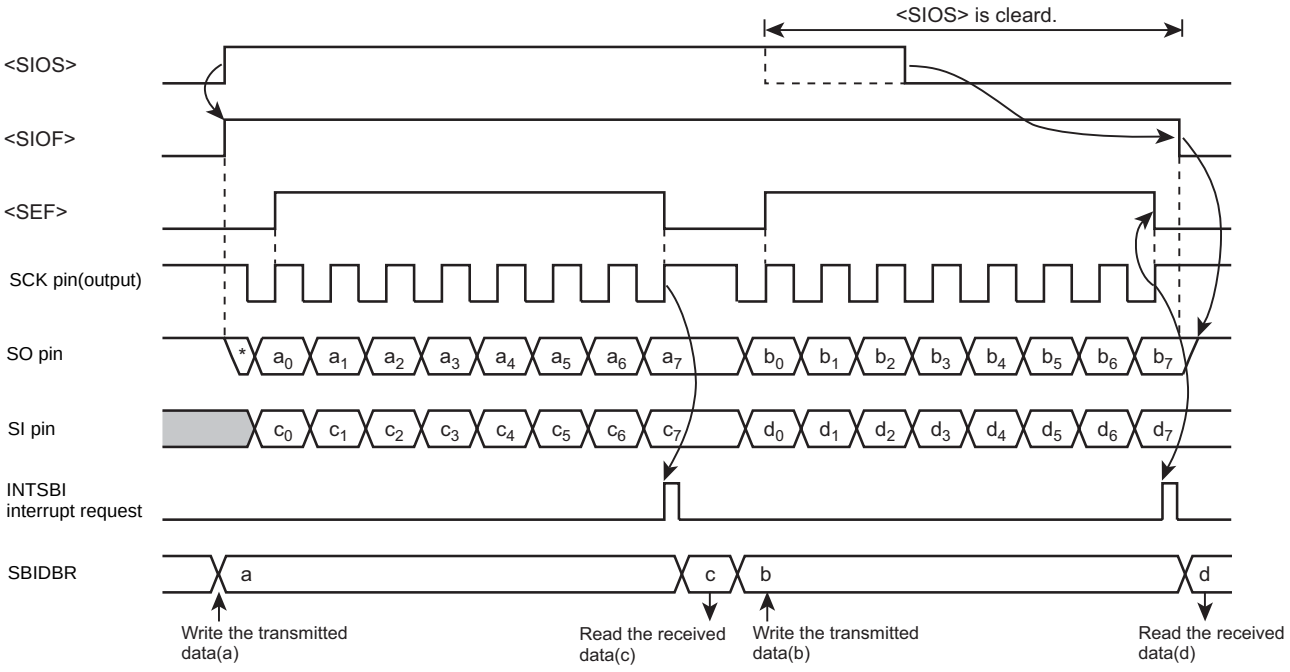


Figure 10-20 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBICR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBICR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBI interrupt

Reg.	←	SBIDBR								Reads the received data.
SBIDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

10.8.2.4 Data retention time of the last bit at the end of transmission

Under the condition SBICR1<SIOS>= "0", the last bit of the transmitted data retains the data of SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

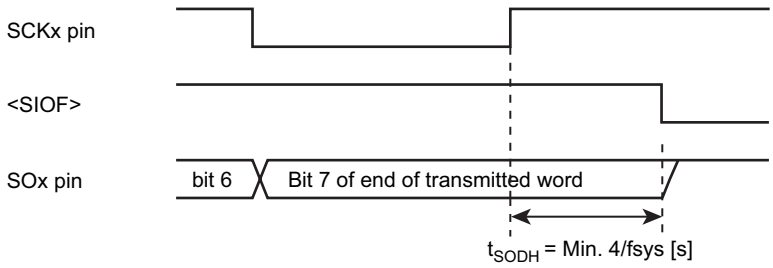


Figure 10-21 Data retention time of the last bit at the end of transmission

11. 12-Bit Analog-to-Digital Converters

The TMPM376FDDFG/FDFG contains two 12-bit successive-approximation analog-to-digital converters (ADCs).

The ADC unit A (ADC A) has 12 analog inputs. Three inputs are able to use for shunt resistor currents of motor 0. Twelve inputs can use for external input.

The ADC unit B (ADC B) has 13 analog inputs. Three inputs are able to use for shunt resistor currents of motor 0. And an input is able to use for shunt resistor currents of motor 1. Thus thirteen inputs can use for external input.

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

11.1 Functions and features

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

11.2 Block Diagram

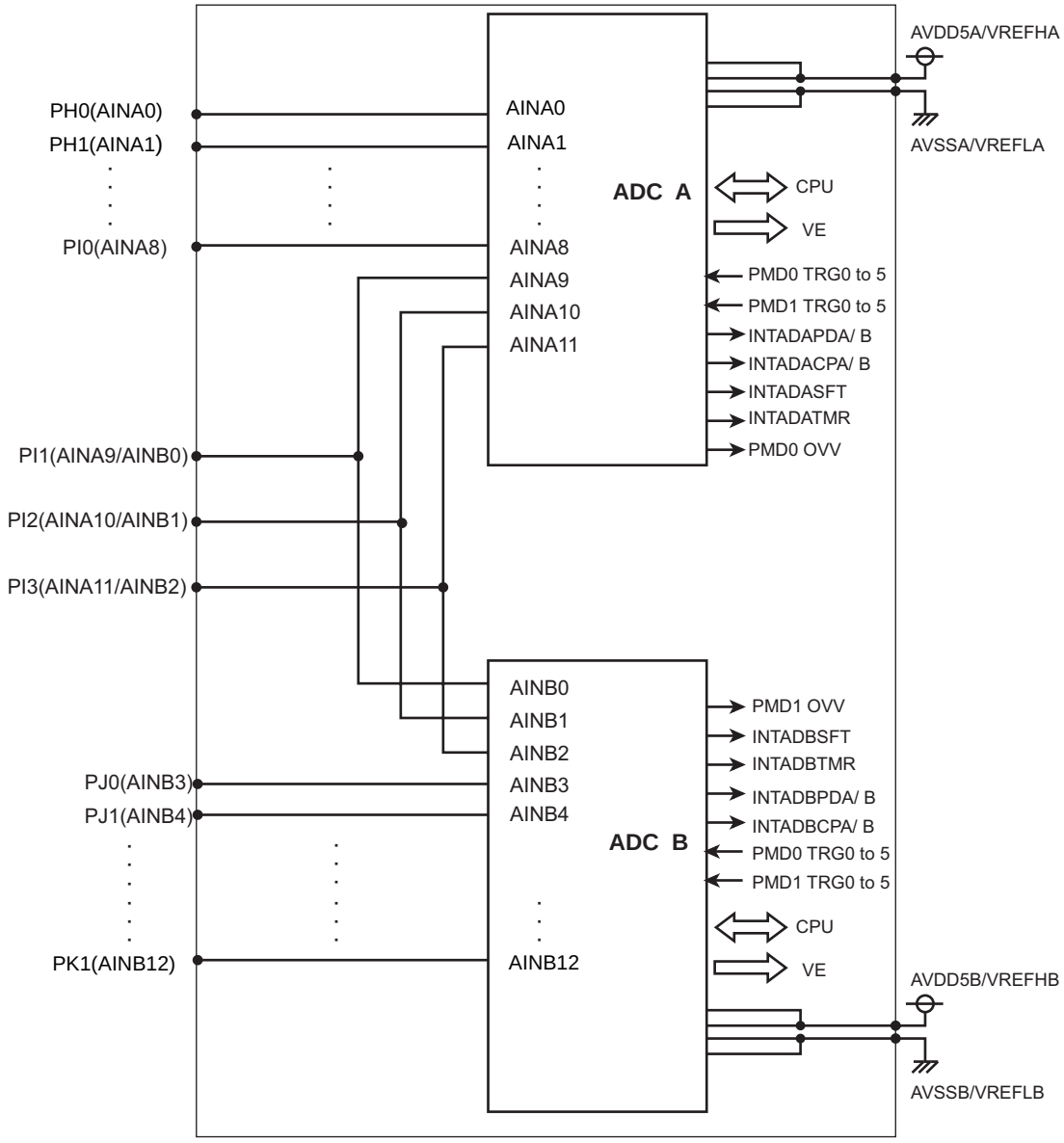


Figure 11-1 AD converters Block Diagram

11.3 List of Registers

Unit x	Base Address
Unit A	0x4003_0000
Unit B	0x4003_0200

Register Name(x=A,B)		Address(Base+)
Clock Setting Register	ADxCLK	0x0000
Mode Setting Register 0	ADxMOD0	0x0004
Mode Setting Register 1	ADxMOD1	0x0008
Mode Setting Register 2	ADxMOD2	0x000C
Monitoring Setting Register 0	ADxCMPCR0	0x0010
Monitoring Setting Register 1	ADxCMPCR1	0x0014
Conversion Result Compare Register 0	ADxCMP0	0x0018
Conversion Result Compare Register 1	ADxCMP1	0x001C
Conversion Result Register 0	ADxREG0	0x0020
Conversion Result Register 1	ADxREG1	0x0024
Conversion Result Register 2	ADxREG2	0x0028
Conversion Result Register 3	ADxREG3	0x002C
Conversion Result Register 4	ADxREG4	0x0030
Conversion Result Register 5	ADxREG5	0x0034
Conversion Result Register 6	ADxREG6	0x0038
Conversion Result Register 7	ADxREG7	0x003C
Conversion Result Register 8	ADxREG8	0x0040
Conversion Result Register 9	ADxREG9	0x0044
Conversion Result Register 10	ADxREG10	0x0048
Conversion Result Register 11	ADxREG11	0x004C
PMD Trigger Program Number Select Register 0	ADxPSEL0	0x0050
PMD Trigger Program Number Select Register 1	ADxPSEL1	0x0054
PMD Trigger Program Number Select Register 2	ADxPSEL2	0x0058
PMD Trigger Program Number Select Register 3	ADxPSEL3	0x005C
PMD Trigger Program Number Select Register 4	ADxPSEL4	0x0060
PMD Trigger Program Number Select Register 5	ADxPSEL5	0x0064
PMD Trigger Program Number Select Register 6	ADxPSEL6	0x0068
PMD Trigger Program Number Select Register 7	ADxPSEL7	0x006C
PMD Trigger Program Number Select Register 8	ADxPSEL8	0x0070
PMD Trigger Program Number Select Register 9	ADxPSEL9	0x0074
PMD Trigger Program Number Select Register 10	ADxPSEL10	0x0078
PMD Trigger Program Number Select Register 11	ADxPSEL11	0x007C
PMD Trigger Interrupt Select Register 0	ADxPINTS0	0x0080
PMD Trigger Interrupt Select Register 1	ADxPINTS1	0x0084

Register Name(x=A,B)		Address(Base+)
PMD Trigger Interrupt Select Register 2	ADxPINTS2	0x0088
PMD Trigger Interrupt Select Register 3	ADxPINTS3	0x008C
PMD Trigger Interrupt Select Register 4	ADxPINTS4	0x0090
PMD Trigger Interrupt Select Register 5	ADxPINTS5	0x0094
PMD Trigger Program Register 0	ADxPSET0	0x0098
PMD Trigger Program Register 1	ADxPSET1	0x009C
PMD Trigger Program Register 2	ADxPSET2	0x00A0
PMD Trigger Program Register 3	ADxPSET3	0x00A4
PMD Trigger Program Register 4	ADxPSET4	0x00A8
PMD Trigger Program Register 5	ADxPSET5	0x00AC
Timer Trigger Program Registers 0 to 3	ADxTSET03	0x00B0
Timer Trigger Program Registers 4 to 7	ADxTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADxTSET811	0x00B8
Software Trigger Program Registers 0 to 3	ADxSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADxSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADxSSET811	0x00C4
Constant Conversion Program Registers0 to 3	ADxASET03	0x00C8
Constant Conversion Program Registers 4 to 7	ADxASET47	0x00CC
Constant Conversion Program Registers 8 to 11	ADxASET811	0x00D0
Mode Setting Register 3	ADxMOD3	0x00D4

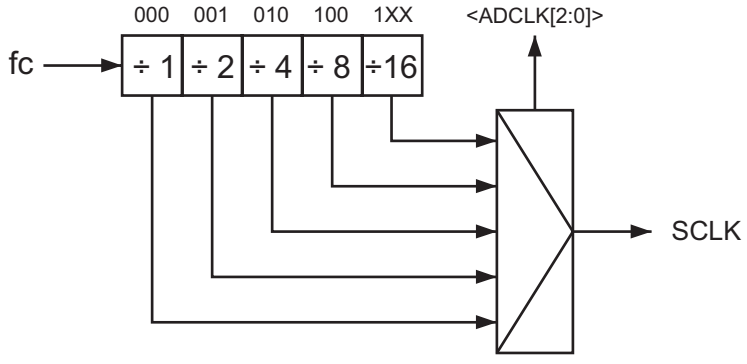
11.4 Register Descriptions

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

11.4.1 ADxCLK (Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-7	–	R	Read as "0".
6-3	TSH[3:0]	R/W	Write as "1001".
2-0	ADCLK[2:0]	R/W	AD prescaler output (SCLK) select 000: fc (Note1) 001: fc/2 010: fc/4 011: fc/8 1xx: fc/16



Note 1: Frequency of SCLK can be use up to 40MHz. Do not set <ADCLK[2:0]> to "000" when fc > 40MHz.

Note 2: AD conversion is performed at the clock frequency selected in this register. The conversion clock frequency must be selected to ensure the guaranteed accuracy.

Note 3: The conversion clock must not be changed while AD conversion is in progress.

11.4.2 ADxMOD0 (Mode Setting Register 0)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as "0".
1	DACON	R/W	DAC control 0: OFF 1: ON Setting <DACON> to "1", when using the ADC.
0	ADSS	W	Software triggered conversion 0: Don't care 1: Start Setting <ADSS> to "1" starts AD conversion (software triggered conversion). Receiving trigger signal from PMD or TMRB(interrupt) starts AD conversion also. For detail setting, please read the chapter about PMD and TMRB.

11.4.3 ADxMOD1 (Mode Setting Register 1)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as "0".
7	ADEN	R/W	AD conversion control 0: Disable 1: Enable Setting <ADEN> to "1", when using the ADC. After Setting <ADEN> to "1", setting <ADAS> to "1" starts AD conversion and repeat conversion.
6-1	–	R	Read as "0".
0	ADAS	R/W	Constant AD conversion control 0: Disable 1: Enable

11.4.4 ADxMOD2 (Mode Setting Register 2)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as "0".
1	ADSFN	R	Software conversion busy flag 0: Conversion completed 1: Conversion in progress The <ADSFN> is a software AD conversion busy flag. After <ADSS> was set to "1", when AD conversion is actually started, <ADSFN> is set to "1". When finished AD conversion, <ADSFN> is cleared to "0".
0	ADBFN	R	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress The <ADBFN> is an AD conversion busy flag. When AD conversion is started regardless of conversion factor (PMD, Timer, Software, Constant), <ADBFN> is set to "1". When finished AD conversion, <ADBFN> is cleared to "0".

11.4.5 ADxMOD3 (Mode Setting Register 3)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-11	–	R/W	Write as "0".
10	–	R/W	Write as "1".
9	–	R/W	Write as "0".
8	–	R/W	Write as "0".
7	–	R/W	Write as "0".
6	–	R/W	Write as "1".
5-3	PMODE[2:0]	R/W	Write as "100".
2-0	–	R/W	Write as "0".

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

11.4.6 ADxCMPCR0(Monitoring Setting Register 0)

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

(n=A,B / A: Monitor0 / B:Monitor1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT0[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons • • 15: After 16 comparisons
7	CMP0EN	R/W	Monitoring function 0:Disable 1:Enable
6-5	-	R	Read as "0".
4	ADBIG0	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register
3-0	REGS0[3:0]	R/W	AD conversion result register to be compared 0000: ADxREG0 / 0100: ADxREG4 / 1000: ADxREG8 0001: ADxREG1 / 0101: ADxREG5 / 1001: ADxREG9 0010: ADxREG2 / 0110: ADxREG6 / 1010: ADxREG10 0011: ADxREG3 / 0111: ADxREG7 / 1011: ADxREG11

Note: 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.

11.4.7 ADxCMPCR1(Monitoring Setting Register 1)

After fixing the conversion result, the interrupt signal (INTADxCPn) is generated.(n=A,B / A:Monitor0 / B:Monitor1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT1[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons • • 15: After 16 comparisons
7	CMP1EN	R/W	Monitoring function 0:Disable 1:Enable
6-5	-	R	Read as "0".
4	ADBIG1	R/W	Comparison condition 0:Larger than or equal to compare register 1:Smaller than or equal to compare register
3-0	REGS1[3:0]	R/W	AD conversion result register to be compared 0000: ADxREG0 0100: ADxREG4 1000: ADxREG8 0001: ADxREG1 0101: ADxREG5 1001: ADxREG9 0010: ADxREG2 0110: ADxREG6 1010: ADxREG10 0011: ADxREG3 0111: ADxREG7 1011: ADxREG11

Note: 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.

11.4.8 ADxCMP0(Conversion Result Compare Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP0				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP0[11:0]	R/W	The value to be compared with an AD conversion result Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

11.4.9 ADxCMP1(Conversion Result Compare Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP1				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP1[11:0]	R/W	The value to be compared with an AD conversion result Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

11.4.10ADxREG0(Conversion Result Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR0				-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR0[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR0	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG0 is read and is cleared when the low-order byte of ADxREG0 is read.
0	ADR0RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR0RF> is a flag that is set when an AD conversion result is stored in the ADxREG0 register and is cleared when the low-order byte of ADxREG0 is read.

11.4.11 ADxREG1(Conversion Result Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR1				-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR1[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR1	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG1 is read and is cleared when the low-order byte of ADxREG1 is read.
0	ADR1RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR1RF> is a flag that is set when an AD conversion result is stored in the ADxREG1 register and is cleared when the low-order byte of ADxREG1 is read.

11.4.12ADxREG2(Conversion Result Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR2				-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR2[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR2	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG2 is read and is cleared when the low-order byte of ADxREG2 is read.
0	ADR2RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR2RF> is a flag that is set when an AD conversion result is stored in the ADxREG2 register and is cleared when the low-order byte of ADxREG2 is read.

11.4.13ADxREG3(Conversion Result Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR3				-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR3[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR3	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG3 is read and is cleared when the low-order byte of ADxREG3 is read.
0	ADR3RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR3RF> is a flag that is set when an AD conversion result is stored in the ADxREG3 register and is cleared when the low-order byte of ADxREG3 is read.

11.4.14ADxREG4(Conversion Result Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR4				-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR4[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR4	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG4 is read and is cleared when the low-order byte of ADxREG4 is read.
0	ADR4RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR4RF> is a flag that is set when an AD conversion result is stored in the ADxREG4 register and is cleared when the low-order byte of ADxREG4 is read.

11.4.15ADxREG5(Conversion Result Register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR5				-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR5[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR5	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG5 is read and is cleared when the low-order byte of ADxREG5 is read.
0	ADR5RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR5RF> is a flag that is set when an AD conversion result is stored in the ADxREG5 register and is cleared when the low-order byte of ADxREG5 is read.

11.4.16ADxREG6(Conversion Result Register 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR6							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR6				-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR6[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR6	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG6 is read and is cleared when the low-order byte of ADxREG6 is read.
0	ADR6RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR6RF> is a flag that is set when an AD conversion result is stored in the ADxREG6 register and is cleared when the low-order byte of ADxREG6 is read.

11.4.17ADxREG7(Conversion Result Register 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR7							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR7				-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR7[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR7	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG7 is read and is cleared when the low-order byte of ADxREG7 is read.
0	ADR7RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR7RF> is a flag that is set when an AD conversion result is stored in the ADxREG7 register and is cleared when the low-order byte of ADxREG7 is read.

11.4.18ADxREG8(Conversion Result Register 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR8							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR8				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR8[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR8	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG8 is read and is cleared when the low-order byte of ADxREG8 is read.
0	ADR8RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR8RF> is a flag that is set when an AD conversion result is stored in the ADxREG8 register and is cleared when the low-order byte of ADxREG8 is read.

11.4.19ADxREG9(Conversion Result Register 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR9							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR9				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR9[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR9	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG9 is read and is cleared when the low-order byte of ADxREG9 is read.
0	ADR9RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR9RF> is a flag that is set when an AD conversion result is stored in the ADxREG9 register and is cleared when the low-order byte of ADxREG9 is read.

11.4.20ADxREG10(Conversion Result Register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR10[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR10	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG10 is read and is cleared when the low-order byte of ADxREG10 is read.
0	ADR10RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR10RF> is a flag that is set when an AD conversion result is stored in the ADxREG10 register and is cleared when the low-order byte of ADxREG10 is read.

11.4.21ADxREG11(Conversion Result Register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR11							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR11				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR11[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR11	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG11 is read and is cleared when the low-order byte of ADxREG11 is read.
0	ADR11RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR11RF> is a flag that is set when an AD conversion result is stored in the ADxREG11 register and is cleared when the low-order byte of ADxREG11 is read.

11.4.22PMD 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.

(x=A,B : ADC Unit)

- PMD Trigger Program Number Select Register (ADxPSEL0 to ADxPSEL11)

  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(PMD0TRG0 to 5 , PMD1TRG0 to 5) generated by the PMD. Programs 0 to 5 are available.

  "ADxPSEL0 to ADxPSEL5" corresponds to "PMD0TRG0 to 5". "ADxPSEL6 to ADxPSEL11" corresponds to "PMD1TRG0 to 5".

- PMD Trigger Interrupt Select Register (ADxPINTS0 to ADxPINTS5)

  The PMD Trigger Interrupt Select Registers (ADxPINTS0 to ADxPINTS5) select the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

  ADxPINTS0 corresponds to program 0, and it exists to ADxPINT5 (program 5).

- PMD Trigger Program Register (ADxPSET0 to ADxPSET5)

  The PMD Trigger Program Setting Registers (ADxPSET0 to ADxPSET5) specify the settings for each of programs 0 to 5. Each PMD Trigger Program Register is comprised of four registers for specifying the AIN input to be converted. The conversion results corresponding to the ADxPSETn0 to ADxPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADxREG0 to ADxREG3).

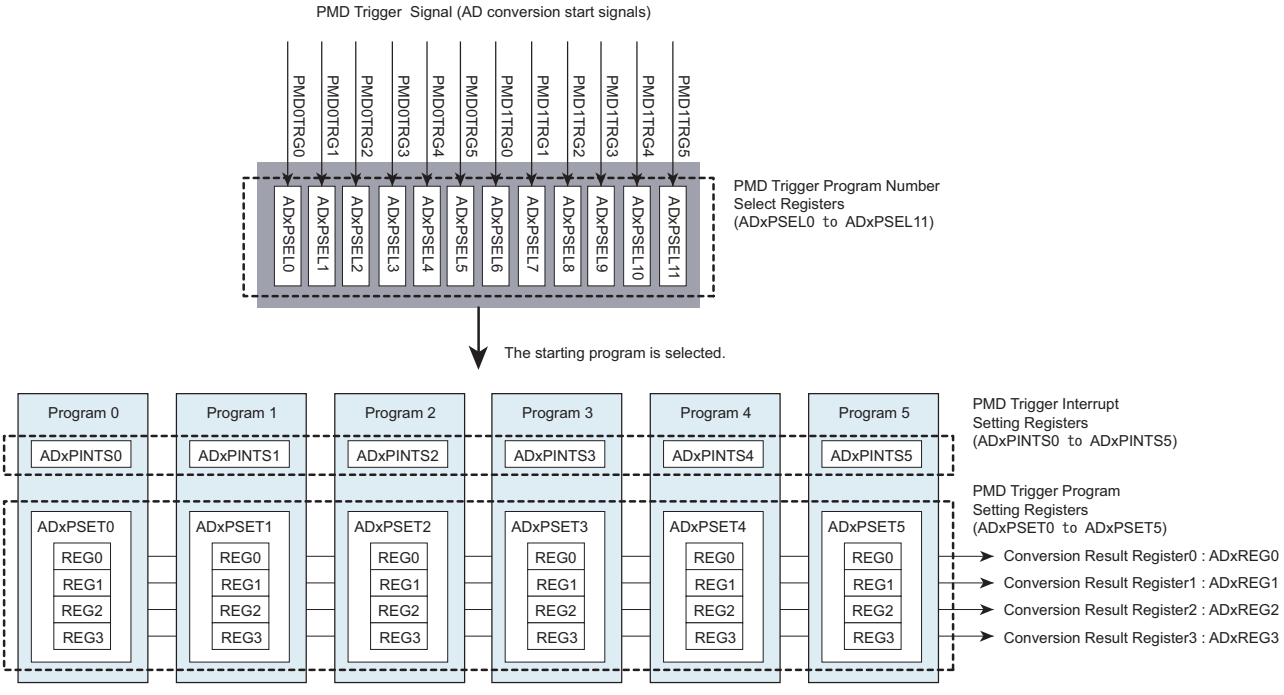


Figure 11-2 PMD Trigger Program Registers

11.4.22.1ADxPSEL0 to ADxPSEL11(PMD Trigger Program Number Select Register 0 to 11)

ADxPSEL0:PMD Trigger Program Number Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS0	-	-	-	-	PMDS0		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS0	R/W	PMD0TRG0 trigger controll 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS0[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL1:PMD Trigger Program Number Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS1	-	-	-	-	PMDS1		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS1	R/W	PMD0TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS1[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL2:PMD Trigger Program Number Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS2	-	-	-	-	PMDS2		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS2	R/W	PMD0TRG2 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS2[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL3:PMD Trigger Program Number Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS3	-	-	-	-	PMDS3		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS3	R/W	PMD0TRG3 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS3[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL4:PMD Trigger Program Number Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS4	-	-	-	-	PMDS4		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS4	R/W	PMD0TRG4 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS4[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL5:PMD Trigger Program Number Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS5	-	-	-	-	PMDS5		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS5	R/W	PMD0TRG5 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS5[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL6:PMD Trigger Program Number Select Register 6

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS6	-	-	-	-	PMDS6		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS6	R/W	PMD1TRG0 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS6[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL7:PMD Trigger Program Number Select Register 7

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS7	-	-	-	-	PMDS7		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS7	R/W	PMD1TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS7[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL8:PMD Trigger Program Number Select Register 8

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS8	-	-	-	-	PMDS8		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS8	R/W	PMD1TRG2 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS8[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL9:PMD Trigger Program Number Select Register 9

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS9	-	-	-	-	PMDS9		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS9	R/W	PMD1TRG3 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS9[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL10:PMD Trigger Program Number Select Register 10

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS10	-	-	-	-	PMDS10		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS10	R/W	PMD1TRG4 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS10[2:0]	R/W	Program number select (Refer to Table 11-1)

ADxPSEL11:PMD Trigger Program Number Select Register 11

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS11	-	-	-	-	PMDS11		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS11	R/W	PMD1TRG5 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS11[2:0]	R/W	Program number select (Refer to Table 11-1)

Table 11-1 Program number select

<PMDS0[2:0]>~ <PMDS11[2:0]>	
000	Program0
001	Program1
010	Program2
011	Program3
100	Program4
101	Program5
110	Reserved
111	Reserved

11.4.22.2ADxPINTS0 to 5(PMD Trigger Interrupt Select Register 0 to 5)

ADxPINTS0:PMD Trigger Interrupt Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL0	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL0[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 0.

ADxPINTS1:PMD Trigger Interrupt Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL1	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL1[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 1.

ADxPINTS2:PMD Trigger Interrupt Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL2	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL2[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 2.

ADxPINTS3:PMD Trigger Interrupt Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL3	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL3[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 3.

ADxPINTS4:PMD Trigger Interrupt Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL4	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL4[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 4.

ADxPINTS5:PMD Trigger Interrupt Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL5	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL5[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 5.

11.4.22.3ADxPSET0 to 5(PMD Trigger Program Register 0 to 5)

Each ADxPSETn (n=0 to 5:Program number) is composed of four sets that assume <AINSPnm [4:0]>, <UVWISnm[1:0]>, and <ENSPnm> in a couple.

(m=0 to 3)(x=A,B : ADC Unit)

ADxREGm ADxPSETn	m=0	m=1	m=2	m=3
n=0	<ENSP00> <UVWIS00> <AINSP00>	<ENSP01> <UVWIS01> <AINSP01>	<ENSP02> <UVWIS02> <AINSP02>	<ENSP03> <UVWIS03> <AINSP03>
n=1	<ENSP10> <UVWIS10> <AINSP10>	<ENSP11> <UVWIS11> <AINSP11>	<ENSP12> <UVWIS12> <AINSP12>	<ENSP13> <UVWIS13> <AINSP13>
n=2	<ENSP20> <UVWIS20> <AINSP20>	<ENSP21> <UVWIS21> <AINSP21>	<ENSP22> <UVWIS22> <AINSP22>	<ENSP23> <UVWIS23> <AINSP23>
n=3	<ENSP30> <UVWIS30> <AINSP30>	<ENSP31> <UVWIS31> <AINSP31>	<ENSP32> <UVWIS32> <AINSP32>	<ENSP33> <UVWIS33> <AINSP33>
n=4	<ENSP40> <UVWIS40> <AINSP40>	<ENSP41> <UVWIS41> <AINSP41>	<ENSP42> <UVWIS42> <AINSP42>	<ENSP43> <UVWIS43> <AINSP43>
n=5	<ENSP50> <UVWIS50> <AINSP50>	<ENSP51> <UVWIS51> <AINSP51>	<ENSP52> <UVWIS52> <AINSP52>	<ENSP53> <UVWIS53> <AINSP53>

Table 11-2 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINA10
0_1011	:AINA11	:AINB11
0_1100	:Reserved	:AINB12
0_1101 to 1_1111	:Reserved	:Reserved

ADxPSET0:PMD Trigger Program Register 0

	31	30	29	28	27	26	25	24
bit symbol	ENSP03	UVWIS03		AINSP03				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP02	UVWIS02		AINSP02				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP01	UVWIS01		AINSP01				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP00	UVWIS00		AINSP00				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP03	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS03[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP03[4:0]	R/W	AIN select Refer to Table 11-2 .
23	ENSP02	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS02[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP02[4:0]	R/W	AIN select Refer to Table 11-2 .
15	ENSP01	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS01[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP01[4:0]	R/W	AIN select Refer to Table 11-2 .
7	ENSP00	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS00[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP00[4:0]	R/W	AIN select Refer to Table 11-2 .

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET1:PMD Trigger Program Register 1

	31	30	29	28	27	26	25	24
bit symbol	ENSP13	UVWIS13		AINSP13				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP12	UVWIS12		AINSP12				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP11	UVWIS11		AINSP11				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP10	UVWIS10		AINSP10				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP13	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS13[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP13[4:0]	R/W	AIN select Refer to Table 11-2 .
23	ENSP12	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS12[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP12[4:0]	R/W	AIN select Refer to Table 11-2 .
15	ENSP11	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS11[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP11[4:0]	R/W	AIN select Refer to Table 11-2 .
7	ENSP10	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS10[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP10[4:0]	R/W	AIN select Refer to Table 11-2 .

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET2:PMD Trigger Program Register 2

	31	30	29	28	27	26	25	24
bit symbol	ENSP23	UVWIS23		AINSP23				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP22	UVWIS22		AINSP22				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP21	UVWIS21		AINSP21				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP20	UVWIS20		AINSP20				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP23	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS23[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP23[4:0]	R/W	AIN select Refer to Table 11-2 .
23	ENSP22	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS22[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP22[4:0]	R/W	AIN select Refer to Table 11-2 .
15	ENSP21	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS21[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP21[4:0]	R/W	AIN select Refer to Table 11-2 .
7	ENSP20	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS20[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP20[4:0]	R/W	AIN select Refer to Table 11-2 .

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET3:PMD Trigger Program Register 3

	31	30	29	28	27	26	25	24
bit symbol	ENSP33	UVWIS33		AINSP33				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP32	UVWIS32		AINSP32				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP31	UVWIS31		AINSP31				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP30	UVWIS30		AINSP30				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP33	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS33[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP33[4:0]	R/W	AIN select Refer to Table 11-2 .
23	ENSP32	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS32[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP32[4:0]	R/W	AIN select Refer to Table 11-2 .
15	ENSP31	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS31[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP31[4:0]	R/W	AIN select Refer to Table 11-2 .
7	ENSP30	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS30[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP30[4:0]	R/W	AIN select Refer to Table 11-2 .

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET4:PMD Trigger Program Register 4

	31	30	29	28	27	26	25	24
bit symbol	ENSP43	UVWIS43		AINSP43				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP42	UVWIS42		AINSP42				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP41	UVWIS41		AINSP41				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP40	UVWIS40		AINSP40				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP43	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS43[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP43[4:0]	R/W	AIN select Refer to Table 11-2 .
23	ENSP42	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS42[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP42[4:0]	R/W	AIN select Refer to Table 11-2 .
15	ENSP41	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS41[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP41[4:0]	R/W	AIN select Refer to Table 11-2 .
7	ENSP40	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS40[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP40[4:0]	R/W	AIN select Refer to Table 11-2 .

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET5:PMD Trigger Program Register 5

	31	30	29	28	27	26	25	24
bit symbol	ENSP53	UVWIS53		AINSP53				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP52	UVWIS52		AINSP52				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP51	UVWIS51		AINSP51				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP50	UVWIS50		AINSP50				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP53	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS53[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP53[4:0]	R/W	AIN select Refer to Table 11-2 .
23	ENSP52	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS52[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP52[4:0]	R/W	AIN select Refer to Table 11-2 .
15	ENSP51	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS51[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP51[4:0]	R/W	AIN select Refer to Table 11-2 .
7	ENSP50	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS50[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP50[4:0]	R/W	AIN select Refer to Table 11-2 .

Phase select

00	Not specified
01	U
10	V
11	W

11.4.23ADxTSET03 / ADxTSET47 / ADxTSET811 (Timer Trigger Program Registers)

AD conversion can be started by INTTB51 generated from Timer5(TMRB5) as a trigger. There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to "1" enables the ADxTSETm register. The <AINSTm[4:0]> are used to select the AIN pin to be used. The numbers of the Timer Trigger Program Registers correspond to those of the AD Conversion Result Registers. When finished this AD conversion, interrupt : INTADxTMR is generated.

(m=0 to 11),(x=A,B : ADC Unit)

Table 11-3 Select the AIN pin

<AINST0 [4:0]> to <AINST11 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINA10
0_1011	:AINA11	:AINB11
0_1100	:Reserved	:AINB12
0_1101 to 1_1111	:Reserved	:Reserved

ADxTSET03: Timer Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST3[4:0]	R/W	AIN select Refer to Table 11-3 .
23	ENST2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST2[4:0]	R/W	AIN select Refer to Table 11-3 .
15	ENST1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST1[4:0]	R/W	AIN select Refer to Table 11-3 .
7	ENST0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST0[4:0]	R/W	AIN select Refer to Table 11-3 .

ADxTSET47: Timer Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST6	-	-	AINST6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST5	-	-	AINST5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST4	-	-	AINST4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST7[4:0]	R/W	AIN select Refer to Table 11-3 .
23	ENST6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST6[4:0]	R/W	AIN select Refer to Table 11-3 .
15	ENST5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST5[4:0]	R/W	AIN select Refer to Table 11-3 .
7	ENST4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST4[4:0]	R/W	AIN select Refer to Table 11-3 .

ADxTSET811: Timer Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST10	-	-	AINST10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST9	-	-	AINST9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST8	-	-	AINST8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST11[4:0]	R/W	AIN select Refer to Table 11-3 .
23	ENST10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST10[4:0]	R/W	AIN select Refer to Table 11-3 .
15	ENST9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST9[4:0]	R/W	AIN select Refer to Table 11-3 .
7	ENST8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST8[4:0]	R/W	AIN select Refer to Table 11-3 .

11.4.24ADxSSET03 / ADxSSET47 / ADxSSET811(Software Trigger Program Registers)

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. Setting the <ENSSm> to "1" enables the ADxSSETm register. The <AINSSm[4:0]> are used to select the AIN pin to be used. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers. When finished this AD conversion, interrupt :INTADxSFT is generated.

(m=0 to 11),(x=A,B : ADC Unit)

Table 11-4 Select the AIN pin

<AINSS0 [4:0]> to <AINSS11 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINA10
0_1011	:AINA11	:AINB11
0_1100	:Reserved	:AINB12
0_1101 to 1_1111	:Reserved	

ADxSSET03: Software Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS2	-	-	AINSS2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS1	-	-	AINSS1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS0	-	-	AINSS0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS3[4:0]	R/W	AIN select Refer to Table 11-4 .
23	ENSS2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS2[4:0]	R/W	AIN select Refer to Table 11-4 .
15	ENSS1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS1[4:0]	R/W	AIN select Refer to Table 11-4 .
7	ENSS0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS0[4:0]	R/W	AIN select Refer to Table 11-4 .

ADxSSET47: Software Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS6	-	-	AINSS6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS5	-	-	AINSS5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS4	-	-	AINSS4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS7[4:0]	R/W	AIN select Refer to Table 11-4 .
23	ENSS6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS6[4:0]	R/W	AIN select Refer to Table 11-4 .
15	ENSS5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS5[4:0]	R/W	AIN select Refer to Table 11-4 .
7	ENSS4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS4[4:0]	R/W	AIN select Refer to Table 11-4 .

ADxSSET811: Software Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS10	-	-	AINSS10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS9	-	-	AINSS9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS8	-	-	AINSS8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS11[4:0]	R/W	AIN select Refer to Table 11-4 .
23	ENSS10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS10[4:0]	R/W	AIN select Refer to Table 11-4 .
15	ENSS9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS9[4:0]	R/W	AIN select Refer to Table 11-4 .
7	ENSS8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS8[4:0]	R/W	AIN select Refer to Table 11-4 .

11.4.25ADxASET03 / ADxASET47 / ADxASET811(Constant Conversion Program Registers)

The ADCs allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to "1" enables the ADxASETm register. The <AINSAm[4:0]> are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m=0 to 11),(x=A,B : ADC Unit)

Table 11-5 Select the AIN pin

<AINSA0 [4:0]> to <AINSA11 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINA10
0_1011	:AINA11	:AINB11
0_1100	:Reserved	:AINB12
0_1101 to 1_1111	:Reserved	

ADxASET03: Constant Conversion Program Registers03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA2	-	-	AINSA2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA1	-	-	AINSA1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA0	-	-	AINSA0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA3[4:0]	R/W	AIN select Refer to Table 11-5 .
23	ENSA2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA2[4:0]	R/W	AIN select Refer to Table 11-5 .
15	ENSA1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA1[4:0]	R/W	AIN select Refer to Table 11-5 .
7	ENSA0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA0[4:0]	R/W	AIN select Refer to Table 11-5 .

ADxASET47: Constant Conversion Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA6	-	-	AINSA6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA5	-	-	AINSA5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA4	-	-	AINSA4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA7[4:0]	R/W	AIN select Refer to Table 11-5 .
23	ENSA6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA6[4:0]	R/W	AIN select Refer to Table 11-5 .
15	ENSA5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA5[4:0]	R/W	AIN select Refer to Table 11-5 .
7	ENSA4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA4[4:0]	R/W	AIN select Refer to Table 11-5 .

ADxASET811: Cnstant Conversion Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSA11	-	-	AINSA11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA10	-	-	AINSA10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA9	-	-	AINSA9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA8	-	-	AINSA8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA11[4:0]	R/W	AIN select Refer to Table 11-5 .
23	ENSA10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA10[4:0]	R/W	AIN select Refer to Table 11-5 .
15	ENSA9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA9[4:0]	R/W	AIN select Refer to Table 11-5 .
7	ENSA8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA8[4:0]	R/W	AIN select Refer to Table 11-5 .

11.5 Operation Descriptions

11.5.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the VREFHA and VREFLA pins are used in ADC A and the VREFHB and VREFLB pins are used in ADC B. There are no registers for controlling current between VREFHA and VREFLA(or, between VREFHB and VREFLB). Inputs to these pins are fixed.

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.

11.5.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

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.

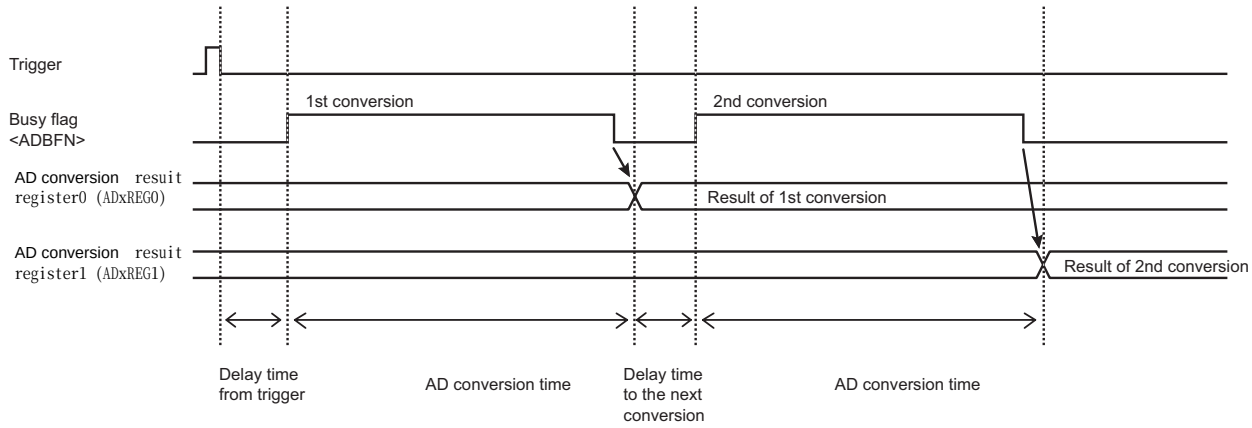


Figure 11-3 Timing chart of AD conversion

Table 11-6 AD conversion time (SCLK = 40MHz)

		fsys = 80MHz		fsys = 40MHz	
	Trigger	MIN	MAX	MIN	MAX
Delay time from trigger [μs] (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[μs]	–	1.85		1.85	
Delay time to the next conversion[μs] (Note2)	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.

11.5.3 AD Conversion Monitoring Function

The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function, set ADxCMPCR0<CMP0EN> or ADxCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by 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.

Note 1: The AD conversion result store flag (<ADR0RF> to <ADR11RF>) 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 (<OVR0> to <OVR11>) is set.

11.6 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

11.6.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADxSSET03, ADxSSET47 and ADxSSET811.(Figure 11-4)

If the ADxMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.(Figure 11-5)

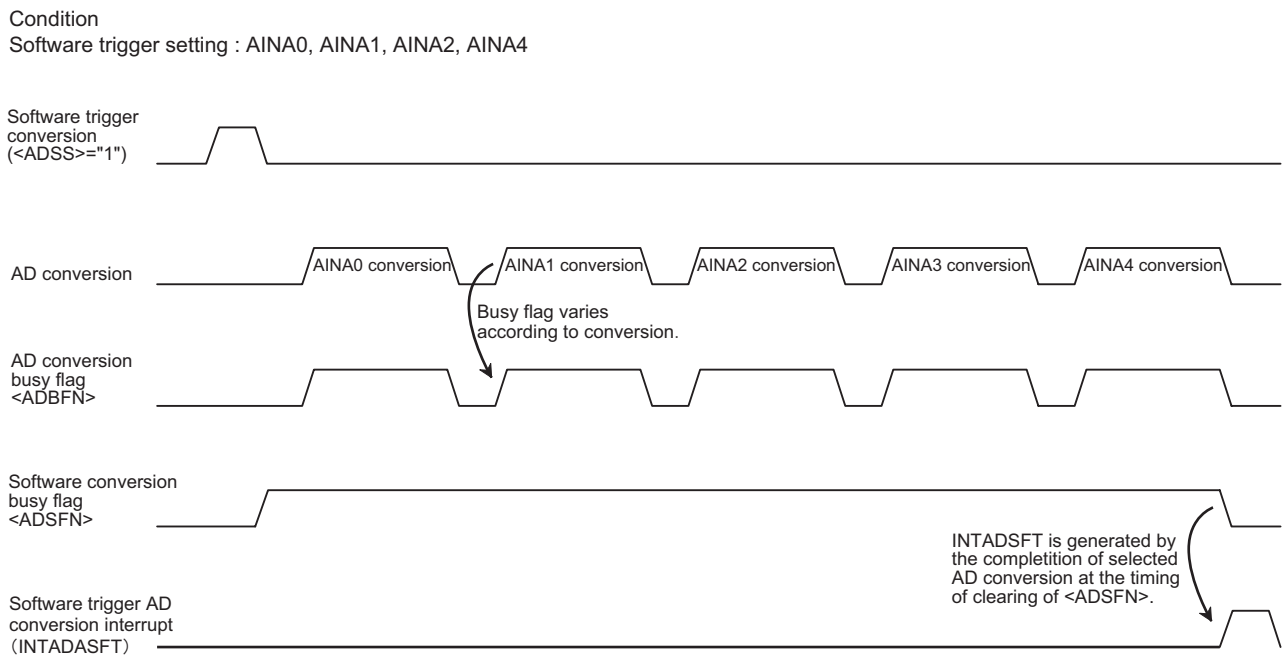


Figure 11-4 Software trigger AD conversion

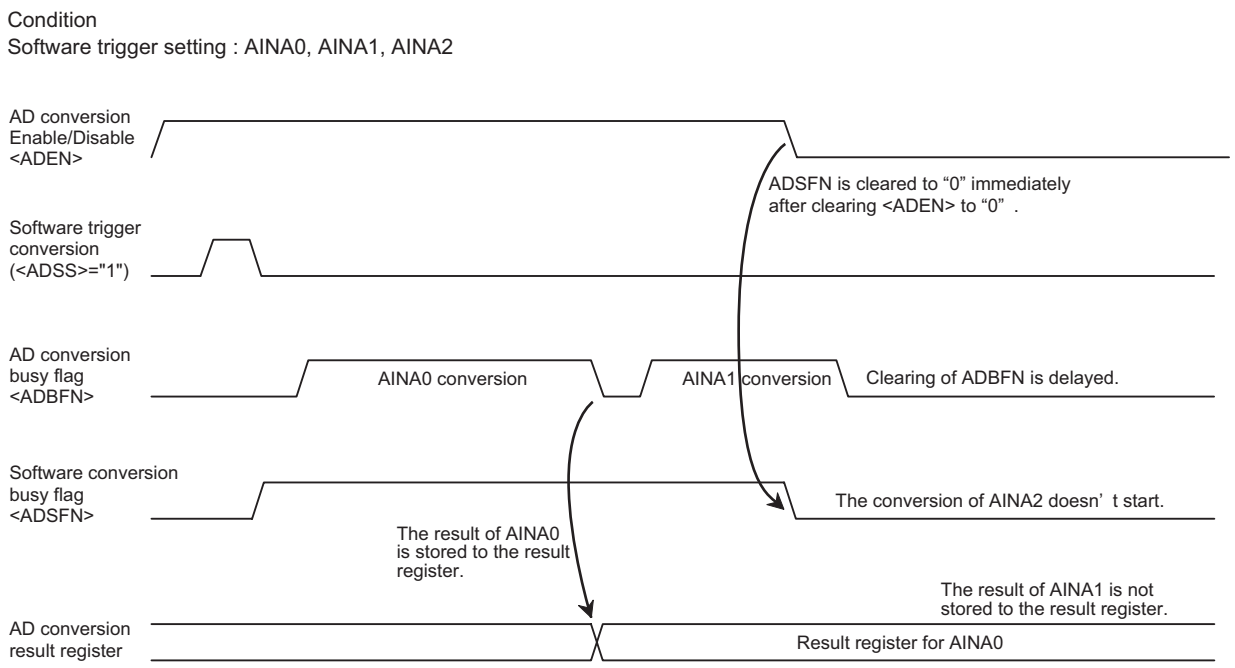


Figure 11-5 Writing "0" to <ADEN> during the software trigger AD conversion

11.6.2 Constant Conversion

In the constant conversion, if the next conversion completes without reading the previous result from the conversion result register, the overrun flag is set to "1". In this case, the previous conversion result in the conversion result register is overwritten by the next result. The overrun flag is cleared by reading of the conversion result.(Figure 11-6)

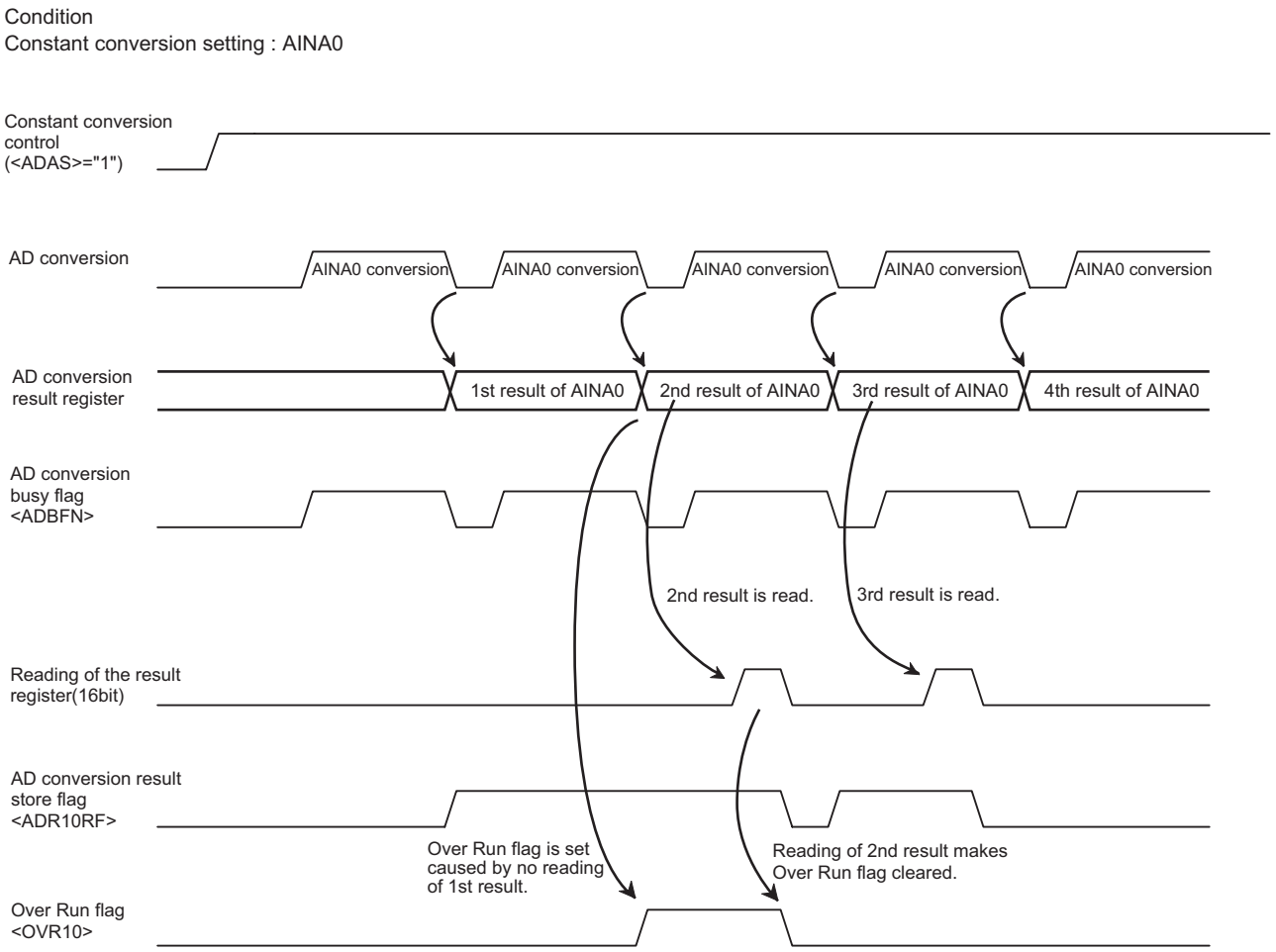


Figure 11-6 Constant conversion

11.6.3 AD conversion by trigger

If the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately.(Figure 11-7) If the timer trigger is occurred during the software trigger conversion, the ongoing conversion stops after the completion of ongoing conversion. (Figure 11-8) After the completion of conversion by trigger, the software trigger conversion starts from the beginning programmed by ADxSSET03, ADxSSET47 and ADxSSET811.(Figure 11-9)

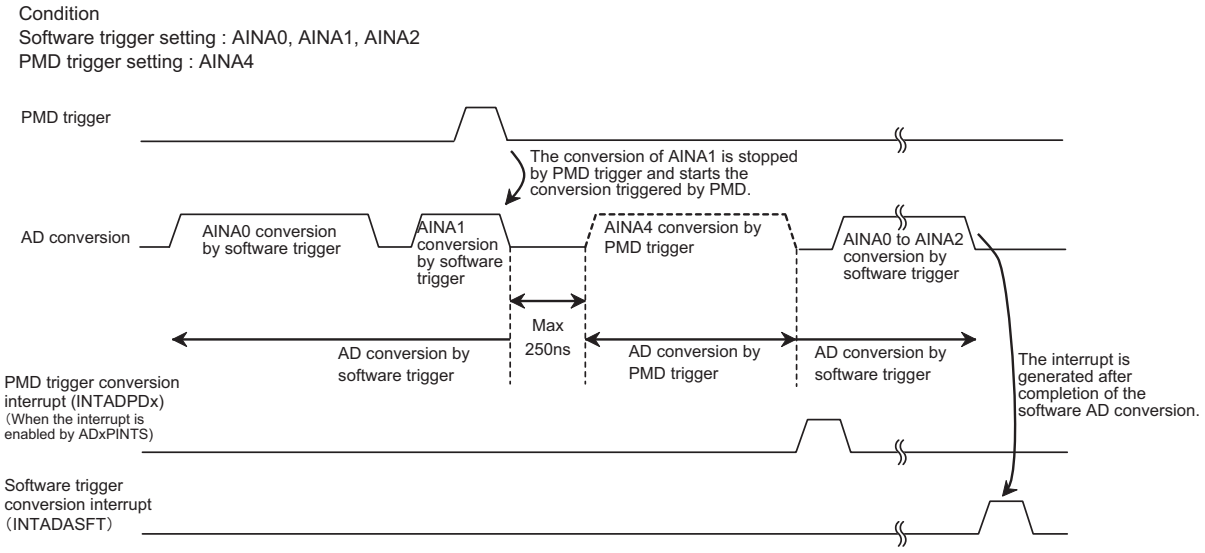


Figure 11-7 AD conversion by PMD trigger

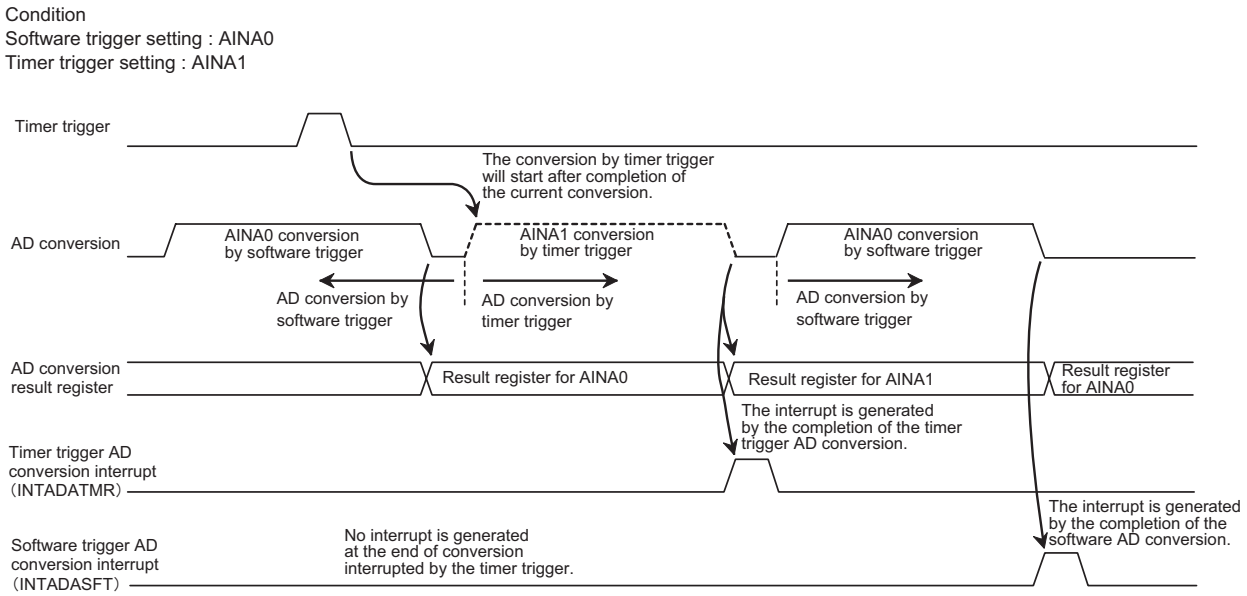


Figure 11-8 AD conversion by timer trigger (1)

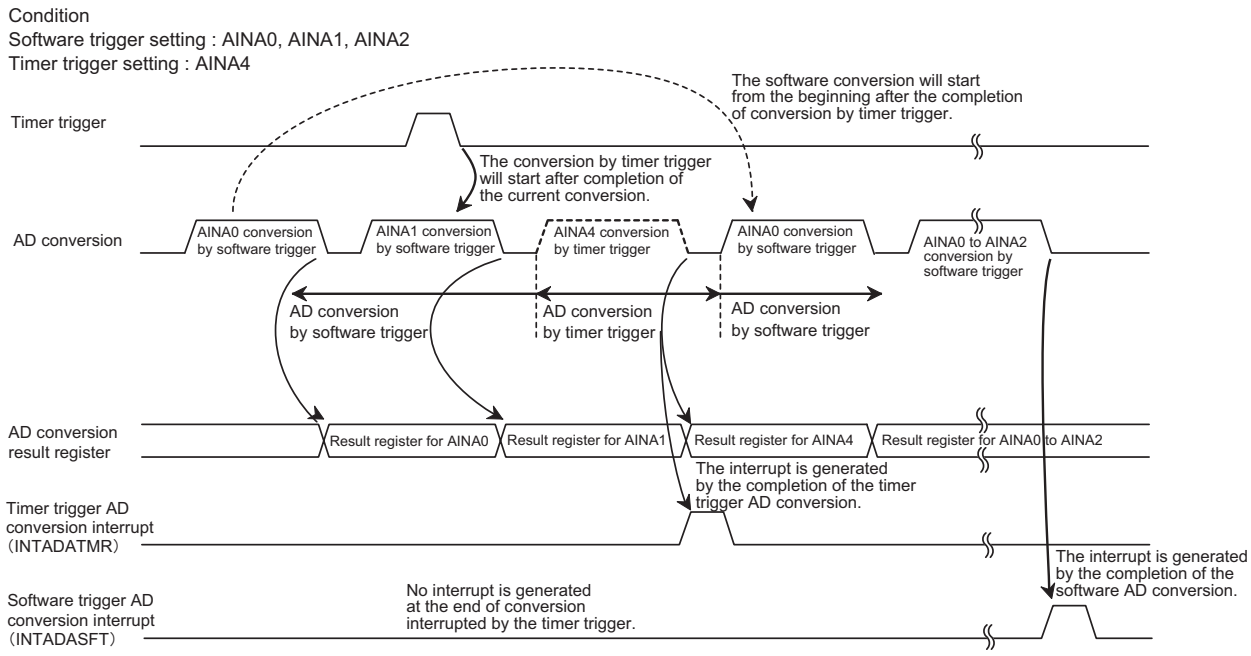
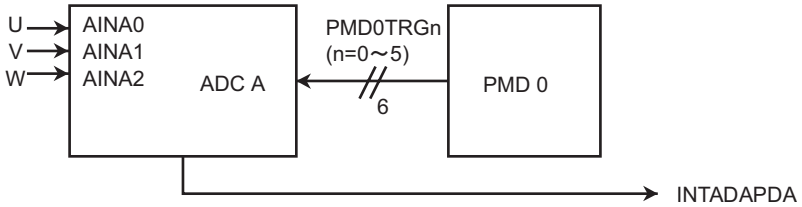


Figure 11-9 AD conversion by timer trigger (2)

11.7 Usage Examples

11.7.1 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.

ADC UnitA

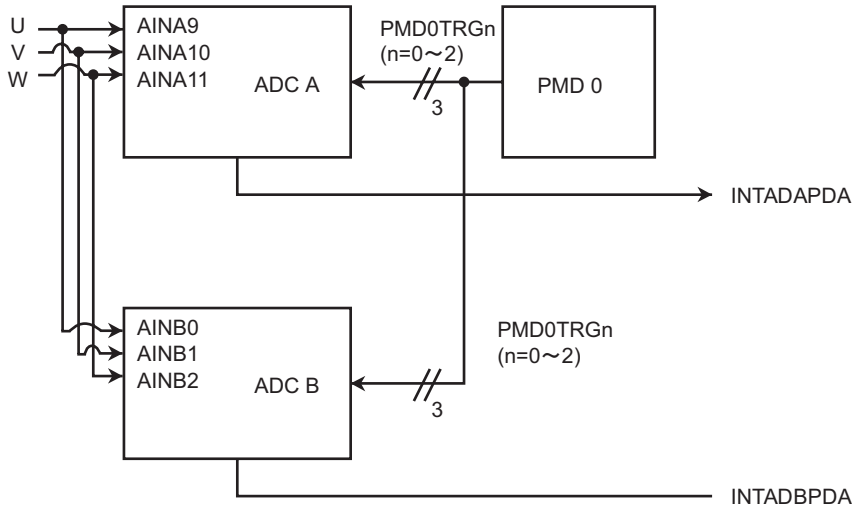
Program	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	A	A	A	A	A	A

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.

11.7.2 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 UnitA

Program	0	1	2
reg0	U	V	W
INT	A	A	A

ADC UnitB

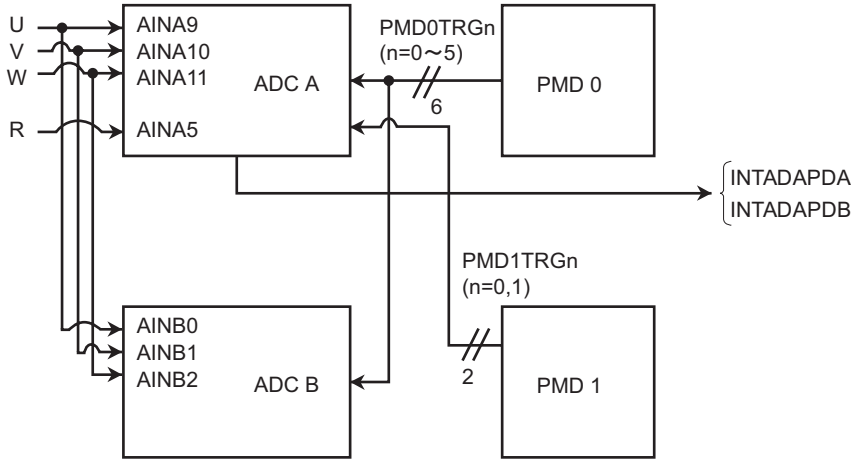
Program	0	1	2
reg0	U	V	W
INT	A	A	A

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.

11.7.3 Successive Conversion Using PMD0 (Three Shunts), PMD1 (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 UnitA

Trigger	PMD0	PMD0	PMD0	PMD1	PMD1
	0,3	1,4	2,5	6	7
Program	0	1	2	3	4
reg0	U	V	W	-	-
reg1	-	-	-	R	-
reg2	-	-	-	-	R
INT	A	A	A	-	B

ADC UnitB

Trigger	PMD0	PMD0	PMD0
	0,3	1,4	2,5
Program	0	1	2
reg0	PMD0 V	PMD0 W	PMD0 U
INT	-	-	-

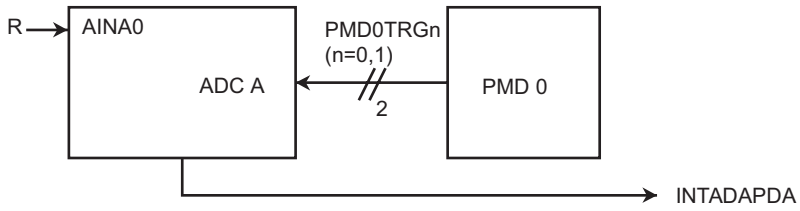
In ADC A, programs 0 to 2 are assigned to six trigger signals from PMD0 and programs 3 and 4 are assigned to two trigger signals from PMD1. In ADC B, programs 0 to 2 are assigned to six trigger signals from PMD0.

"reg0", "reg1" and "reg2" indicate the PMD Trigger Program Registers ADxPSETn[7:0], ADxPSETn[15:8] and ADxPSETn[23:16] (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.

11.7.4 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.

ADC UnitA

Trigger	PMD0	PMD0
	0	1
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 ADAPSETn[7:0] and ADAPSETn[15:8]. "R" indicates a resistor, where the AIN input that is connected to that resistor is set.

When a trigger input occurs, the ADC is started to execute programs 0 and 1 sequentially. When program 1 is completed, the interrupt (INTADAPDA) is generated.

12. Motor Control Circuit (PMD: Programmable Motor Driver)

The TMPM376FDDFG/FDFG contains 2 channels 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.

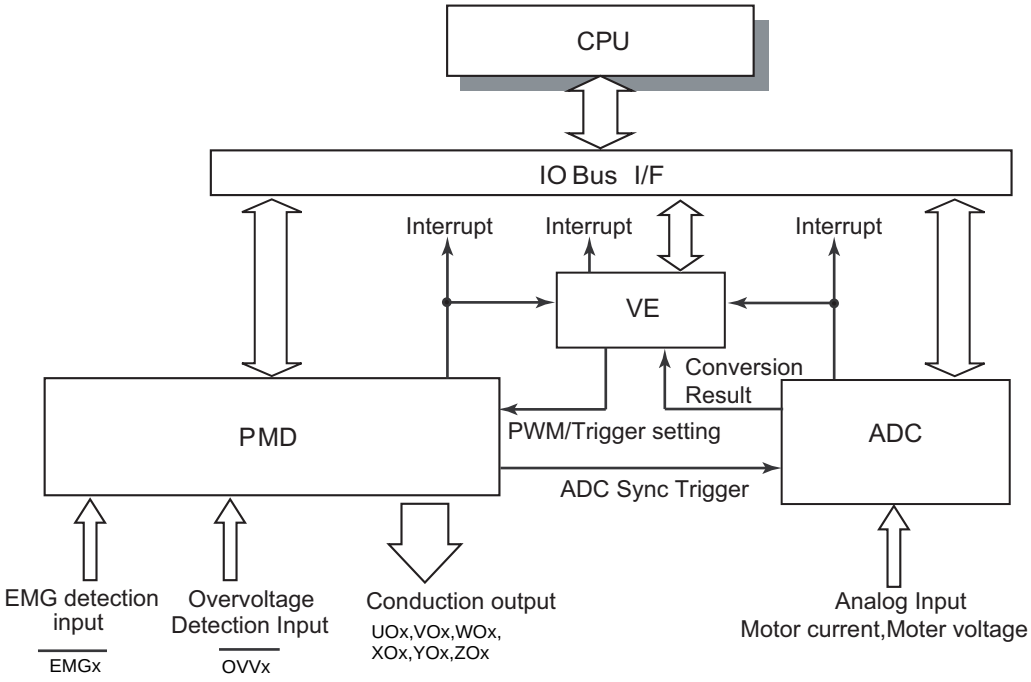


Figure 12-1 Motor Control-related Block Constitution

12.1 PMD Input/Output Signals

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

Table 12-1 Input/Output Signals

Channe	Pin Name	PMD Signal Name	Description
PMD0	PC7/$\overline{OVV0}$	OVV 0	OVV state signal
	PC6/$\overline{EMG0}$	EMG 0	EMG state signal
	PC0/UO0	UO 0	U-phase output
	PC1/XO0	XO 0	X-phase output
	PC2/VO0	VO 0	V-phase output
	PC3/YO0	YO 0	Y-phase output
	PC4/WO0	WO 0	W-phase output
	PC5/ZO0	ZO 0	Z-phase output
PMD1	PG7/$\overline{OVV1}$	OVV 1	OVV state signal
	PG6/$\overline{EMG1}$	EMG 1	EMG state signal
	PG0/UO1	UO 1	U-phase output
	PG1/XO1	XO 1	X-phase output
	PG2/VO1	VO 1	V-phase output
	PG3/YO1	YO 1	Y-phase output
	PG4/WO1	WO 1	W-phase output
	PG5/ZO1	ZO 1	Z-phase output

12.2 PMD Circuit

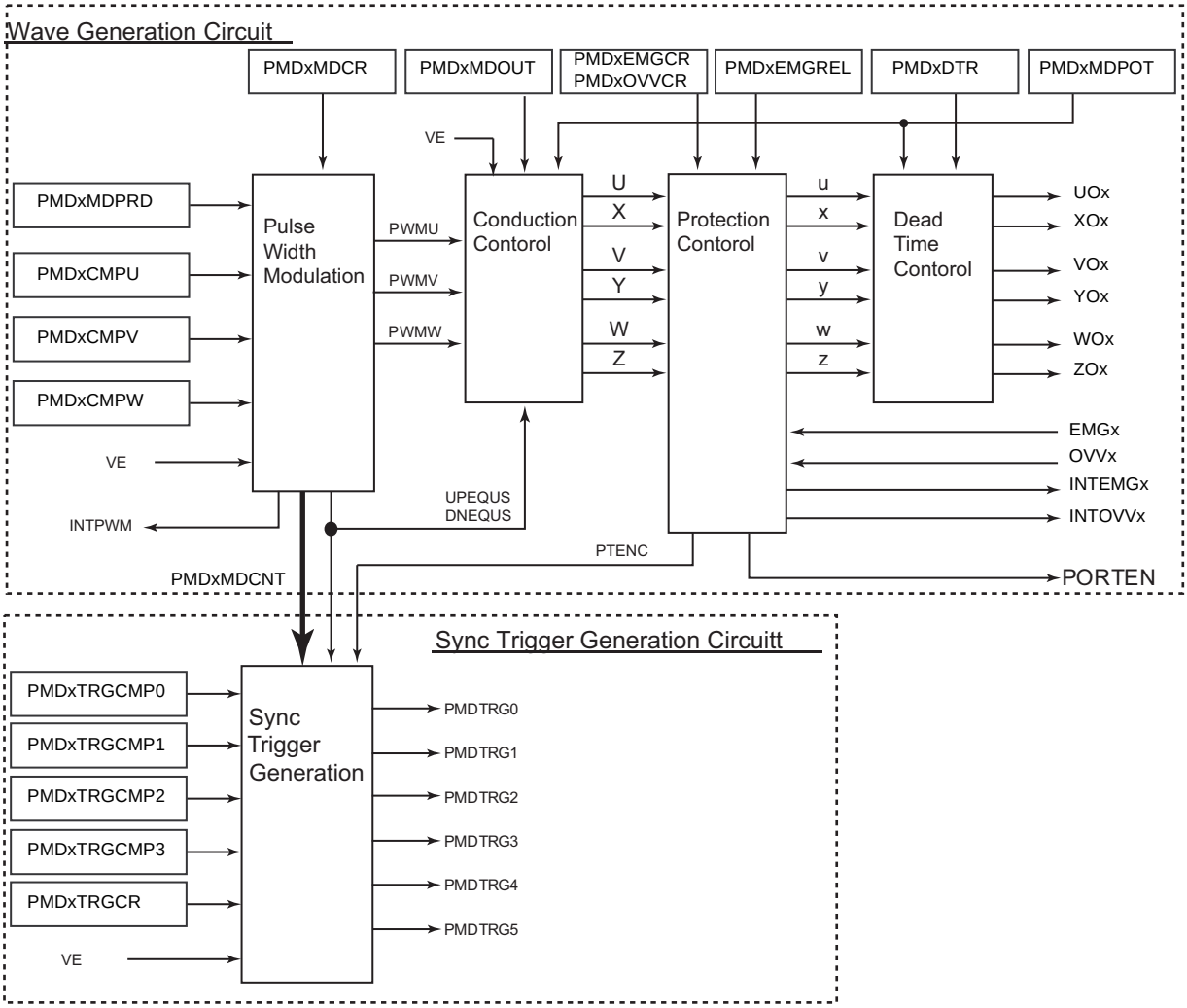


Figure 12-2 Block diagram of PMD Circuit

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.

The table below shows the registers related to the PMD.

12.3 PMD Registers

Channel x	Base Address
Channel 0	0x4005_0400
Channel 1	0x4005_0480

Register Name		Address(Base+)
PMD Enable Register	PMDxMDEN	0x0000
Port Output Mode Register	PMDxPORTMD	0x0004
PMD Control Register	PMDxMDCR	0x0008
PWM Counter Status Register	PMDxCNTSTA	0x000C
PWM Counter Register	PMDxMDCNT	0x0010
PWM Period Register	PMDxMDPRD	0x0014
PMD Compare U Register	PMDxCMPU	0x0018
PMD Compare V Register	PMDxCMPV	0x001C
PMD Compare W Register	PMDxCMPW	0x0020
Mode Select Register	PMDxMODESEL	0x0024
PMD Output Control Register	PMDxMDOUT	0x0028
PMD Output Setting Register	PMDxMDPOT	0x002C
EMG Release Register	PMDxEMGREL	0x0030
EMG Control Register	PMDxEMGCR	0x0034
EMG Status Register	PMDxEMGSTA	0x0038
OVV Control Register	PMDxOVVCR	0x003C
OVV Status Register	PMDxOVVSTA	0x0040
Dead Time Register	PMDxDTR	0x0044
Trigger Compare 0 Register	PMDxTRGCMP0	0x0048
Trigger Compare 1 Register	PMDxTRGCMP1	0x004C
Trigger Compare 2 Register	PMDxTRGCMP2	0x0050
Trigger Compare 3 Register	PMDxTRGCMP3	0x0054
Trigger Control Register	PMDxTRGCR	0x0058
Trigger Output Mode Setting Register	PMDxTRGMD	0x005C
Trigger Output Select Register	PMDxTRGSEL	0x0060
Reserved	–	0x007C

Note: Do not access to "Reserved" address.

12.3.1 PMDxMDEN(PMD Enable Register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	PWMEN	R/W	Enables or disables waveform synthesis. 0: Disable 1: Enable Output ports that are used for the PMD become High-z when the PMD is disabled. Before enabling the PMD, Setting <PWMEN>="1"(enable) other relevant settings, such as output port polarity.

12.3.2 PMDxPORTMD(Port Output Mode Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1-0	PORTMD[1:0]	R/W	Port control setting 00: Upper phases = High-z / lower phases = High-z 01: Upper phases = High-z / ower phases = PMD output 10: Upper phases = PMD output / lower phases = High-z 11: Upper phases = PMD output / lower phases = PMD output The <PORTMD[1:0]> 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-z. In other cases, external port outputs depend on PMD outputs.

Note 1: When <PWMEN>=0, output ports are set to High-z regardless of the output port setting.
Note 2: When an EMG input occurs, external port outputs are controlled depending on the PMDxEMGCR<EMGMD[1:0]> setting.

12.3.3 PMDxMODESEL (Mode Select Register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	MDSEL	R/W	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 (PMDxCMPU,PMDxCMPV, PMDxCMPW), trigger compare registers (PMDxTRGCMP0, PMDxTRGCMP1) andPMDx MDOUT register are double-buffered, and the second buffers are loaded in synchronization with the PMD's internal update timing.

12.3.4 Pulse Width Modulation Circuit

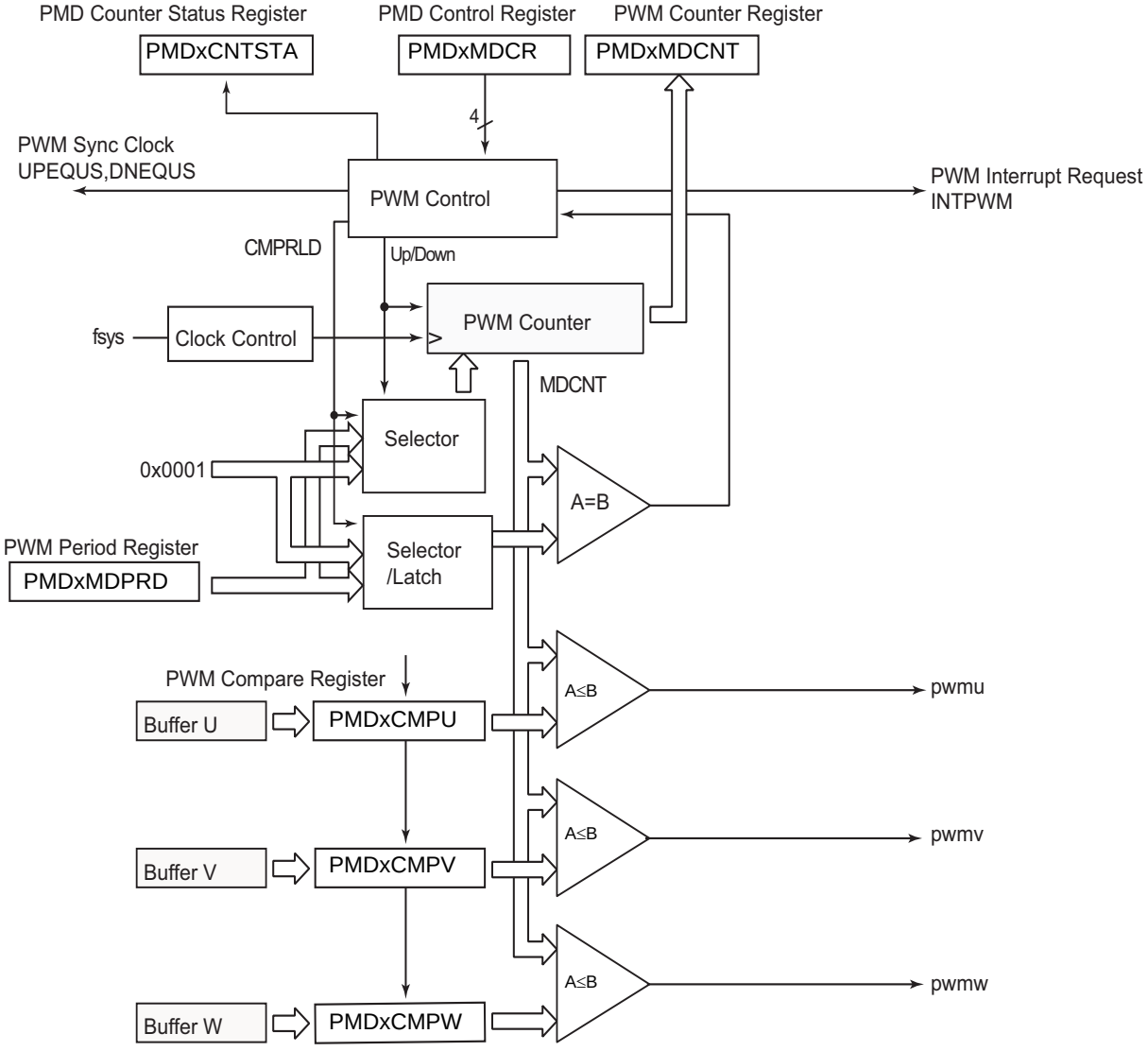


Figure 12-3 Pulse Width Modulation Circuit

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 (PMDxMDCR<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 PMDxMDPRD 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.

$$\text{Sawtooth wave PWM : PMDxMDPRD register Value} = \frac{\text{Oscillation frequency [}}{\text{PWM frequency [Hz]}}$$

$$\text{Trianguler wave PWM : PMDxMDPRD register value} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz] x 2}}$$

2. Compare function

The pulse width modulation circuit compares the PWM compare registers of the 3 phases (PMDx-CMPU / V / W) and the carrier wave generated by the PWM counter (PMDxMDCNT) 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[15:0]> value).

It is also possible to update the compare register at every 0.5 PWM period.

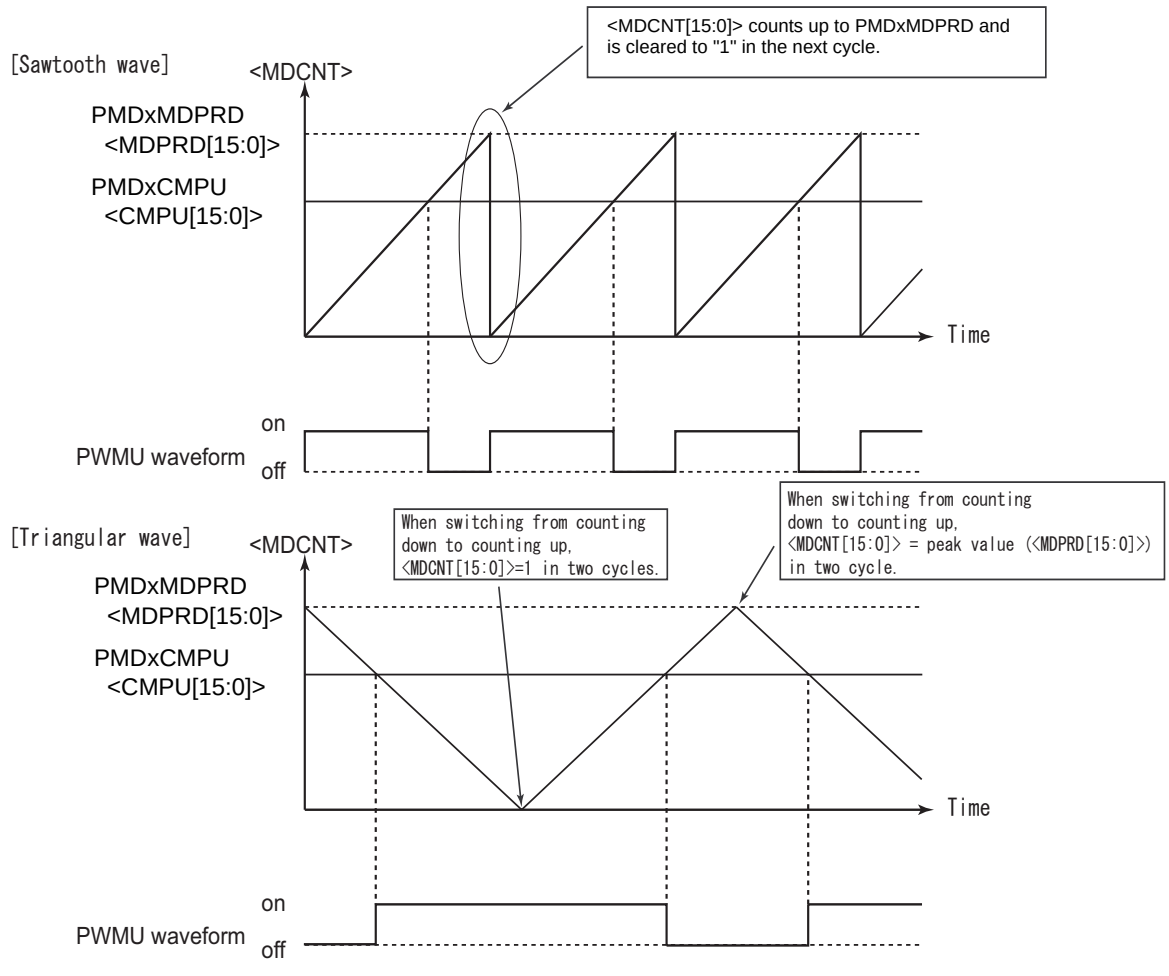


Figure 12-4 PWM Waveforms

3. Waveform mode

Three-phase PWM waveforms can be generated in the following two modes:

1. 3-phase independent mode:

   Each of the PWM compare registers for the three phases is set independently to generate independent PWM waveforms for each phase. This mode is used to generate drive waveforms such as sinusoidal waves.

2. 3-phase common mode:

   Only the U-phase PWM compare register is set to generate identical PWM waveforms for all the three phases. This mode is used for rectangular wave drive of brushless DC motors.

4. Interrupt processing

The pulse width modulation circuit generates PWM interrupt requests in synchronization with PWM waveforms. The PWM interrupt period can be set to half a PWM period, one PWM period, two PWM periods or four PWM periods.

12.3.4.1 PMDxMDCR (PMD Control Register)

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

Bit	Bit Symbol	Type	Function
31-7	-	R	Read as 0.
6	PWMCK	R/W	PWM period extension mode 0: Normal period 1: 4x period When <PWMCK>= "0", the PWM counter operates with a resolution of 12.5 ns at fsys=80 MHz. • Sawtooth wave: 12.5 ns, triangular wave: 25 ns When <PWMCK>="1", the PWM counter operates with a resolution of 50 ns at fsys=80 MHz. • Sawtooth wave: 50 ns, triangular wave: 100 ns
5	SYNTMD	R/W	Port output mode This bit specifies the port output setting of the U, V and W phases. (See Table 12-2.)
4	DTYMD	R/W	Duty mode 0: 3-phase common mode 1: 3-phase independent mode This bit selects whether to make duty setting independently for each phase or to use the PMDxCMPU register for all three phases.
3	PINT	R/W	PWM interrupt timing 0: Interrupt request when PWM counter PMDxMDCNT<MDCNT[15:0]> = 0x0001 1: Interrupt request when PWM counter PMDxMDCNT<MDCNT[15:0]> = <MDPRD[15:0]> 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[15:0]> 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[15:0]>.
2-1	INTPRD[1:0]	R/W	PWM interrupt period 00: Interrupt request at every 0.5 PWM period (<PWMMD>= "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. •note) When <INTPRD[1:0]>= "00", the contents of the compare registers (PMDxCMPU/V/W) and period register (PMDxMDPRD) are updated into their respective buffers when the internal counter equals 1 or the PMDxMDPRD value.
0	PWMMD	R/W	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.

12.3.4.2 PMDxCNTSTA (PWM Counter Status Register)

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

Bit	Bit Symbol	Type	Function
31-1	-	R	Read as 0.
0	UPDWN	R	PWM counter flag 0: Up-counting 1: Down-counting This bit indicates whether the PWM counter is up-counting or down-counting. When the edge-aligned PWM mode is selected, this bit is always read as 0.

12.3.4.3 PMDxMDCNT(PWM Counter Register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	MDCNT[15:0]	R	PWM counter PMD counter value (resolution: 12.5 ns at fsys = 80 MHz) • Sawtooth wave: 12.5 ns, triangular wave: 25 ns • When PMDxMDCR<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 PMDxMDCR<PWMMD>= 0 : 0x0001 In case of PMDxMDCR<PWMMD>= 1 : the value of PMDxMDPRD<MDPRD[15:0]>

12.3.4.4 PMDxMDPRD(PWM Period Register)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	MDPRD[15:0]	R/W	PWM period <MDPRD[15:0]> ≥ 0x010 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[15:0]> value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or <MDPRD[15:0]>. The least significant bit must be set as 0.) If <MDPRD[15:0]> is set to a value less than 0x0010, it is automatically assumed to be 0x0010. (The register retains the actual value that is written.)

Note: **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.**

12.3.4.5 PMDxCMPU (PWM Compare Registers of U Phase)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	CMPUx[15:0]	R/W	PWM pulse width of U Phase Compare registers (resolution : 12.5 ns at fsys =80 MHz) • Sawtooth wave: 12.5 ns, triangular wave: 25 ns • When MDCR<PWMCK>="1", the counter resolution becomes 50 ns. <CMPUx[15:0]> are compare registers for determining the output pulse width of the U phases. Theses registers are double-buffered. Pulse width is determined by comparing the buffer and the PWM counter to evaluate which is larger. (To be loaded when the PWM counter value matches the <MDPRD[15:0]> value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or <MDPRD[15:0]>.) When this register is read, the value of the first buffer (data set via the bus) is returned.

Note 1: **To load the second buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting PMDxMODESEL<MDSEL> to 0.**

Note 2: **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.**

12.3.4.6 PMDxCMPV (PWM Compare Registers of V Phase)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	CMPVx[15:0]	R/W	PWM pulse width of V Phase Compare registers (resolution : 12.5 ns at fsys =80 MHz) • Sawtooth wave: 12.5 ns, triangular wave: 25 ns • When MDCR<PWMCK>=1, the counter resolution becomes 50 ns. <CMPVx[15:0]> are compare registers for determining the output pulse width of the V phases. Theses registers are double-buffered. Pulse width is determined by comparing the buffer and the PWM counter to evaluate which is larger. (To be loaded when the PWM counter value matches the <MDPRD[15:0]> value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or <MDPRD[15:0]>.) When this register is read, the value of the first buffer (data set via the bus) is returned.

Note 1: **To load the second buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting PMDxMODESEL to 0.**

Note 2: **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.**

12.3.4.7 PMDxCMPW (PWM Compare Registers of W Phase)

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

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	CMPWx[15:0]	R/W	PWM pulse width of W Phase Compare registers (resolution : 12.5 ns at fsys =80 MHz) • Sawtooth wave: 12.5 ns, triangular wave: 25 ns • When MDCR<PWMCK>=1, the counter resolution becomes 50 ns. <CMPWx [15:0]> are compare registers for determining the output pulse width of the W phases. Theses registers are double-buffered. Pulse width is determined by comparing the buffer and the PWM counter to evaluate which is larger. (To be loaded when the PWM counter value matches the <MDPRD[15:0]> value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or <MDPRD[15:0]>.) When this register is read, the value of the first buffer (data set via the bus) is returned.

Note 1: **To load the second buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting PMDxMODESEL to 0.**

Note 2: **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.**

12.3.5 Conduction Control Circuit

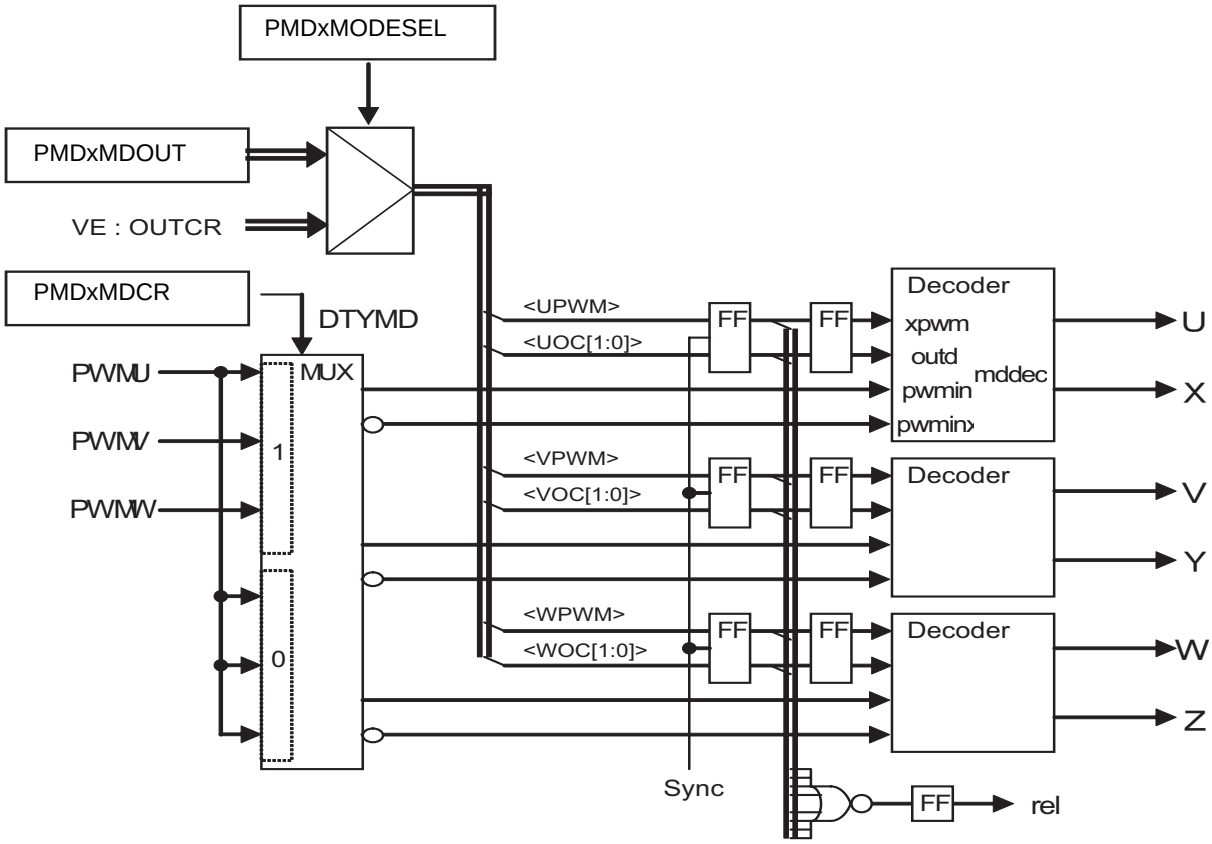


Figure 12-5 Conduction Control Circuit

The conduction control circuit performs output port control according to the settings made in the "PMDxMDOUT". The PMDxMDOUT 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 PMDxMDPOT<POLH><POLL>register and bits 3 and 2 of the PMDxMDPOT register. In addition, bits 10 to 8 of the PMDxMDOUT register select PWM or High/Low output for each of the U, V and W phases. When PWM output is selected, PWM waveforms are output. When High/Low output is selected, output is fixed to either a High or Low level. Table 12-2 shows a summary of port outputs according to port output settings in the PMDxMDOUT register and polarity settings in the PMDxMDCR register.

12.3.5.1 PMDxMDPOT (PMD Output Setting Register)

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

Bit	Bit Symbol	Type	Function
31-4	-	R	Read as 0.
3	POLH	R/W	Upper phase port polarity (Note) 0: Active low 1: Active high POLH selects the output port polarity of the upper phases.
2	POLL	R/W	Lower phase port polarity (Note) 0: Active low 1: Active high POLL selects the output port polarity of the lower phases.
1-0	PSYNCS[1:0]	R/W	MDOUT transfer timing (Note) 00: Async to PWM 01: Load when PWM counter <MDCNT[15:0]> = 1 10: Load when PWM counter <MOCNT[15:0]> = PMDxMDPRD<MDPRD[15:0]> 11: Load when PWM counter <MDCNT[15:0]> = 1 or PMDxMDPRD<MDPRD[15:0]> 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.

Note: This field must be set while PMDxMDEN<PWMEN>=0.

12.3.5.2 PMDxMDOUT(PMD Output Control Register)

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

Bit	Bit Symbol	Type	Function
31-11	-	R	Read as 0.
10	WPWM	R/W	U-, V-, and W-phase output control 0: High/Low Output 1: PWM Output The MDOUT register controls the port outputs of the U, V and W phases (see Table 12-2 below.)
9	VPWM	R/W	
8	UPWM	R/W	
7-6	-	R	Read as 0.
5-4	WOC[1:0]	R/W	U-, V-, and W-phase output control The MDOUT register controls the port outputs of the U, V and W phases (see Table 12-2below.)
3-2	VOC[1:0]	R/W	
1-0	UOC[1:0]	R/W	

Note 1: **To load the second buffer of PWMx MDOUT with a value updated via the bus, select the bus mode (default) by setting PMDxMODESEL to 0.**

Note 2: **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.**

Table 12-2 Port Outputs according to the <UOC>, <VOC>, <WOC>, <UPWM>, <VPWM> and <WPWM> Settings

PMDxMDCR<SYNTMD>=0

Polarity: Active high (PMDxMDPOT<POLH><POLL>="11")

PMDxMDOUT Output Control (Upper phase)	PMDxMDOUT Output Control (Lower phase)	<WPWM><VPWM><UPWM> Output Select			
		0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMDxMDCR<SYNTMD>=0

Polarity: Active low (PMDxMDPOT<POLH><POLL>="00")

PMDxMDOUT Output Control (Upper phase)	PMDxMDOUT Output Control (Lower phase)	<WPWM><VPWM><UPWM> Output Select			
		0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	$\overline{PWM}$
0	1	H	L	H	$\overline{PWM}$
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

PMDxMDCR<SYNTMD>=1

Polarity: Active high (PMDxMDPOT<POLH><POLL>="11")

PMDxMDOUT Output Control (Upper phase)	PMDxMDOUT Output Control (Lower phase)	<WPWM><VPWM><UPWM> Output Select			
		0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	$\overline{PWM}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMDxMDCR<SYNTMD>=1

Polarity: Active low (PMDxMDPOT<POLH><POLL>="00")

PMDxMDOUT Output Control (Upper phase)	PMDxMDOUT Output Control (Lower phase)	<WPWM><VPWM><UPWM> Output Select			
		0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> <UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	$\overline{PWM}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

- Output Settings for One-Shunt MODE

One-Shunt can be supported by the following settings.

Table 12-3 Register Settings for One-Shunt

	Normal PWM center on	U-Phase PWM center off	V-Phase PWM center off	W-Phase PWM center off
CMPU	duty_U	<MDPRD[15:0]>-duty_U	duty_U	duty_U
CMPV	duty_V	duty_V	<MDPRD[15:0]>-duty_V	duty_V
CMPW	duty_W	duty_W	duty_W	<MDPRD[15:0]>-duty_W
<UOC[1:0]>	11	00	11	11
<VOC[1:0]>	11	11	00	11
<WOC[1:0]>	11	11	11	00

12.3.6 Protection Control Circuit

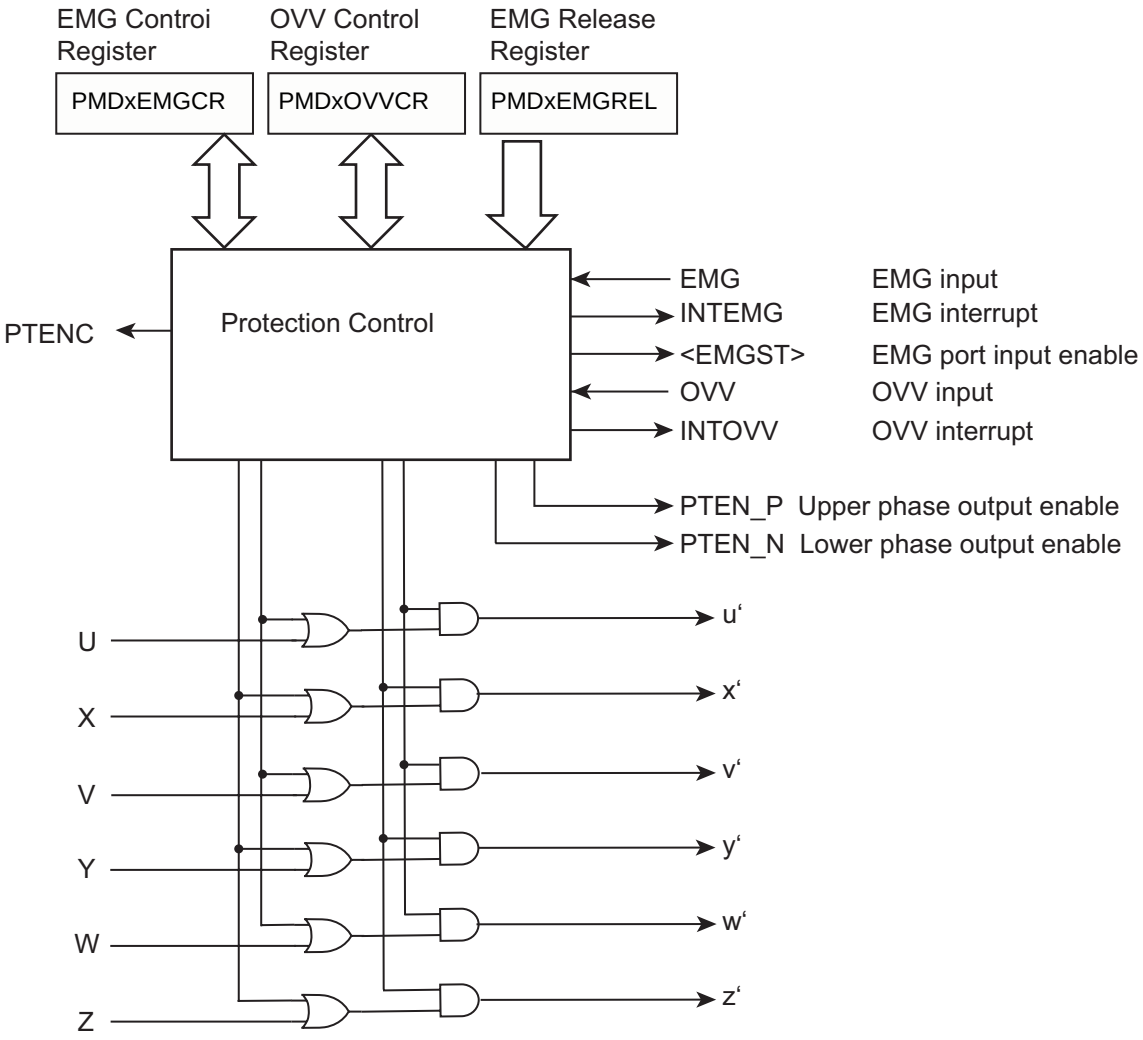


Figure 12-6 Protection Control Circuit

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

12.3.6.1 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 PMDxEMGCR<EMGMD> setting) and an EMG interrupt (INTEMG) is generated. <EMGMD> can be set to output a control signal that sets external output ports to High-z in case of an emergency.

A tool break also disables all six PWM output lines depending on the PMDxPORTMD<PORTMD> setting. When a tool break occurs, external output ports can be set to High-z through the setting of the PMDxEMGSTA<EMGST> register.

EMG protection is set through the EMG Control Register (PMDxEMGCR).

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 (PMDxMDOUT<[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 PMDxMGCR<EMGRS> to 1 to release EMG protection, make sure that PMDxEMGST<EMGI> is high.

The EMG protection circuit can be disabled only after the specified key codes ("0x5A", "0xA5") are written in the <EMGREL> register to prevent it from being inadvertently disabled.

Note: Initial procedure for EMG function

After reset, the EMG function is enabled but EMG pin is configured as a normal port. Therefore, as the EMG protection might be valid, release the EMG protection by the following procedure at the initial sequence.

1: Selects EMG function by PxFR register.

2: Reads PMDxEMGSTA<EMGI> to confirm it as "1".

3: Sets PMDxMDOUT<[10:8]>, <[5:0]> to "0" to make all ports in-active ("L" output).

4: Releases EMG protection by setting PMDxEMGCR<EMGRS> to "1".

If the EMG protection is to be disabled, continue the following procedure.

5: Writes the key codes to PMDxEMGREL (In order of "0x5A" and "0xA5")

6: Sets PMDxEMGCR<EMGEN> to "0" to disable the EMG protection.

12.3.6.2 PMDxEMGREL (EMG Release Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	EMGREL[7:0]	W	EMG disable code The EMG and OVV protection functions can be disabled by setting 0x5A and 0xA5 in this order to bits 7 to 0 of the <EMGREL[7:0]> register. When disabling these functions, <EMGEN> and <OVVEN> must be cleared to "0". • This register is used for both the EMG and OVV functions.

12.3.6.3 PMDxEMGCR (EMG Control Register)

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

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as 0.
11-8	EMGCNT[3:0]	R/W	EMG input detection time The noise remove time value can be calculated by following formula. <EMGCNT[3:0]> × 16/fsys (resolution: 200[nsec] at 80 MHz) <EMGCNT[3:0]>= 0 to 15 (When <EMGCNT[3:0]> = 0, the noise filter is bypassed.)
7-6	-	R	Read as 0.
5	INHEN	R/W	Tool break enable/disable 0: Disable 1: Enable This bit selects whether or not to stop the PMD when the PMD stop signal is input from the tool. In the initial state, tool breaks are enabled.
4-3	EMGMD[1:0]	R/W	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.
2	-	R/W	Read as "0".
1	EMGRS	W	EMG protection release 0: - 1: Release protection EMG protection can be released by setting the PMDxMDOUT register to 0 and then setting the <EMGRS> bit to 1. This bit is always read as 0. • PMDxMDOUT register 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 PMDxEMGSTA<EMGI> has returned to High.
0	EMGEN	R/W	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 0x5A and 0xA5 in this order to the PMDxEMGREL<EMGREL> register and then clear the EMGEN bit to 0. (These three instructions must be executed consecutively.)

12.3.6.4 PMDxEMGSTA (EMG Status Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1	EMGI	R	EMG input EMG protection state The EMG input state can be known by reading this bit
0	EMGST	R	EMG protection state 0: Normal operation 1: Protected The EMG protection state can be known by reading this bit.

12.3.6.5 OVV Protection Control Circuit (OVV Block)

The OVV protection control circuit consists of an OVV protection control unit and a port output disable unit. This circuit is activated when the OVV input port is asserted.

When the OVV input is asserted (H→L) for a specified period (set in OVVCR<OVVCNT>), the OVV protection circuit fixes the six port output lines in the conduction control circuit to high or low. At this time, an OVV interrupt (INTOVV) is generated.

It is possible to turn off only the upper or lower phases or all phases.

OVV protection is set through the "PMDxOVVCR". A read value of "1" in PMDxOVVSTA<OVVST> indicates that the OVV protection circuit is active.

The release of the OVV protection state is enabled by setting PMDxOVVCR<OVVRS> 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 PMDxOVVSTA<OVVI>.)

The OVV protection state is released in synchronization with the PWM period (when the PWM count matches the <MDPRD[15:0]> value. When 0.5 PWM period is selected, the release timing is when the PWM counter equals 1 or <MDPRD[15:0]>.). To disable the OVV protection function, write "0x5A" and "0xA5" in this order to the <EMGREL[7:0]> and then clear PMDxOVVCR<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[7:0]> register to prevent it from being inadvertently disabled.

12.3.6.6 PMDxOVVCR (OVV Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	-	-	-	-	OVVCNT			
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	-	ADIN1EN	ADIN0EN	OVVMD		OVVISEL	OVVRS	OVVEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as 0.
11-8	OVVCNT[3:0]	R/W	OVV input detection time OVVCNT = 1 to 15 (If 0 is set, it is handled as 1.) OVVCNT × 16/fsys (resolution: 200[nsec] at 80 MHz) • OVVCNT is effective only when port input is selected as the OVV signal (<OVVISEL> = "1").
7	-	R	Read as 0.
6	ADIN1EN	R/W	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.
5	ADIN0EN	R/W	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.
4-3	OVVMD[1:0]	R/W	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 <EMGMD[1:0]> register become effective.
2	OVVISEL	R/W	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[3:0]> becomes invalid.
1	OVVRS	R/W	OVV protection release 0: Disable automatic release of OVV protection 1: Enable 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[15:0]>) register by setting this bit to "1". • When 0.5 PWM period is selected (PMDxMDCR<INTPRD[1:0]> = "00"), the OVV protection state is released when the PWM counter equals "1" or <MDPRD[15:0]> .
0	OVVEN	R/W	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 "0x5A" and "0xA5" in this order to the <EMGREL[7:0]> register and then clear <OVVEN> bit to "0". (These three instructions must be executed consecutively.)

12.3.6.7 PMDxOVVSTA (OVV Status Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1	OVVI	R	OVVI input OVVI state The OVV input state (selected by OVVCR<OVVISEL>) can be known by reading this bit.
0	OVVST	R	OVV protection state 0: Normal operation 1: Protected The OVV state can be known by reading this bit.

12.3.7 Dead Time Circuit

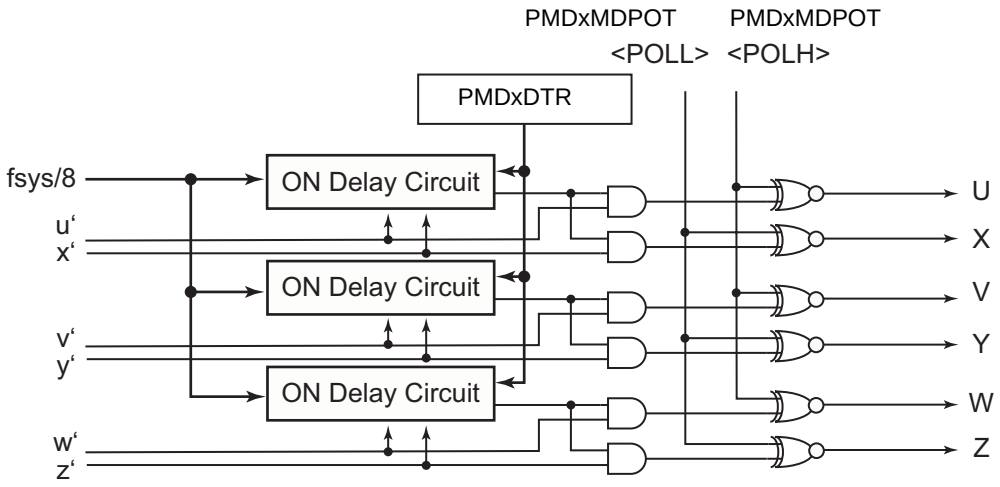


Figure 12-7 Dead Time Circuit

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 (PMDxDTR<DTR[7:0]>)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 PMDxMDPOT<POLH> and <POLL>.

12.3.7.1 PMDxDTR (Dead Time Register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DTR[7:0]	R/W	Dead time The Dead time value can be calculated by following formula. 100 nsec × <DTR[7:0]> (up to 25.5 μsec at fsys = 80 MHz)

Note: **Do not change <DTR[7:0]> register while PMDxMDEN<PWMEN> = 1.**

12.3.8 Sync Trigger Generation Circuit

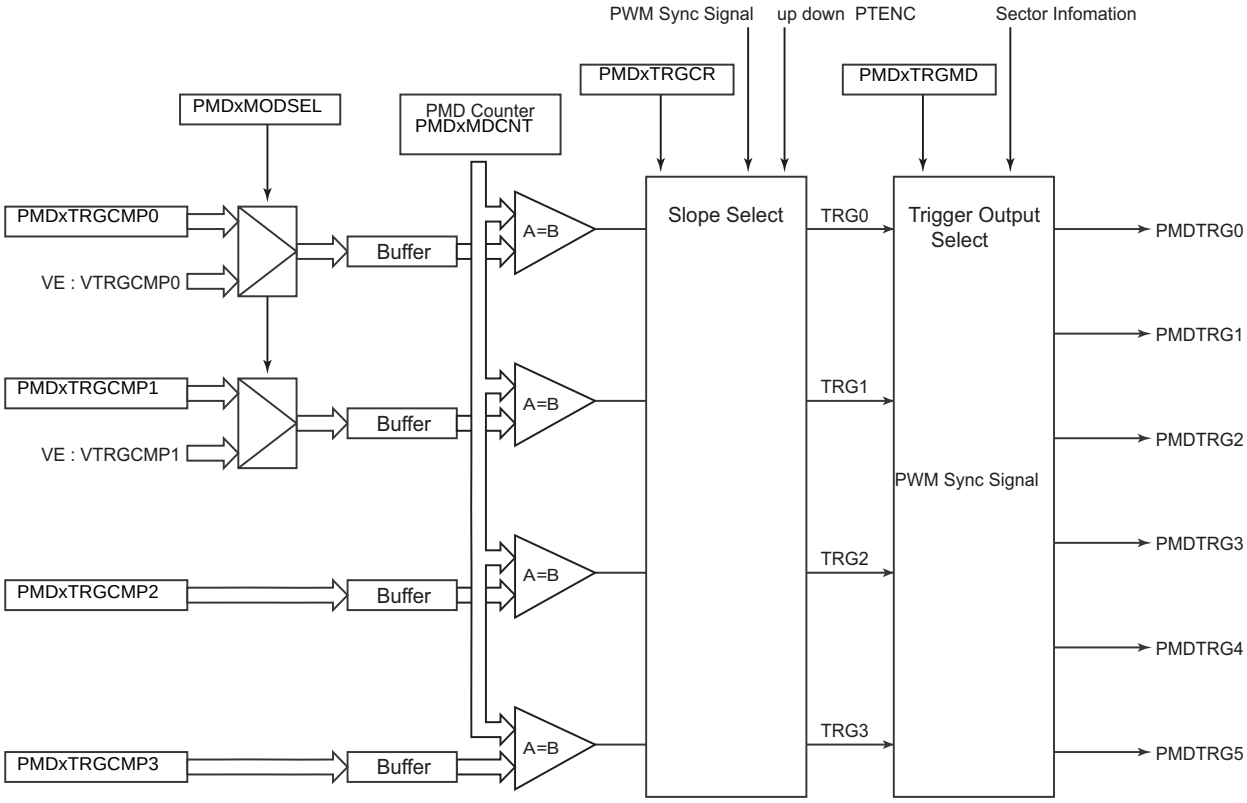


Figure 12-8 Sync Trigger Generation Circuit

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 PMDxMDCNT and PMDxTRGCMP. 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 (PMDxMDEN<PWMEN> = 0), trigger output is also disabled.

When the trigger select output mode is selected, the trigger output port is switched according to the PMDxTRGSEL<TRGSEL> register setting or sector information from the Vector Engine.

12.3.8.1 PMDxTRGCMP0 (Trigger Compare Registers 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TRGCMP0							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TRGCMP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP0 [15:0]	R/W	Trigger output compare registers When the PMD counter value <MDCNT[15:0]> matches the value set in TRGCMP0, PMDTRG is output. When TRGCMP0 is read, the value in the first buffer of the double buffers (data set via the bus) is returned. TRGCMP0 should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. It is prohibited to set <TRGCMP0> to 0 or the <MDPRD[15:0]> value.

Note 1: **To load the data in TRGCMP0 and TRGCMP1 to the second buffers, select the bus mode (default) by setting PMDxMODESEL<MDSEL> to "0".**

Note 2: **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.**

Note 3: **When <TRGCMP0> is set to 0x0001, no trigger output is made only in the first cycle after PWM start (<PWMEN> = 1).**

12.3.8.2 PMDxTRGCMP1 (Trigger Compare Registers1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TRGCMP1							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TRGCMP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP1 [15:0]	R/W	Trigger output compare registers When the PMD counter value <MDCNT[15:0]> matches the value set in TRGCMP1, PMDTRG is output. When TRGCMP1 is read, the value in the first buffer of the double buffers (data set via the bus) is returned. TRGCMP1 should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. It is prohibited to set <TRGCMP1> to 0 or the <MDPRD[15:0]> value.

Note 1: **To load the data in TRGCMP0 and TRGCMP1 to the second buffers, select the bus mode (default) by setting MODESEL PMDxMODESEL<MDSEL> to 0.**

Note 2: **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.**

Note 3: **When <TRGCMP1> is set to 0x0001, no trigger output is made only in the first cycle after PWM start (MDEN<PWMEN> = 1).**

12.3.8.3 PMDxTRGCMP2 (Trigger Compare Registers 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TRGCMP2							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TRGCMP2							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP2 [15:0]	R/W	Trigger output compare registers When the PMD counter value <MDCNT[15:0]> matches the value set in TRGCMP2, PMDTRG is output. When TRGCMP2 is read, the value in the first buffer of the double buffers (data set via the bus) is returned. TRGCMP2 should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. It is prohibited to set <TRGCMP2> to 0 or the <MDPRD[15:0]> value.

Note 1: **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.**

Note 2: **When <TRGCMP2> is set to "0x0001", no trigger output is made only in the first cycle after PWM start (MDEN<PWMEN> = "1").**

12.3.8.4 PMDxTRGCMP3 (Trigger Compare Registers 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TRGCMP3							
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TRGCMP3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15-0	TRGCMP3 [15:0]	R/W	Trigger output compare registers When the PMD counter value <MDCNT[15:0]> matches the value set in TRGCMP3, PMDTRG is output. When TRGCMP3 is read, the value in the first buffer of the double buffers (data set via the bus) is returned. TRGCMP3 should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. It is prohibited to set TRGCMP3 to 0 or the <MDPRD[15:0]> value.

Note 1: **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.**

Note 2: **When <TRGCMP3> is set to "0x0001", no trigger output is made only in the first cycle after PWM start (<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 PMDxTRGCR<TRGxMD[2:0]>. When PMDxTRGCR<TRGxBE> is set to "1", data written to TRGCMPx is immediately loaded to the second buffer.

Table 12-4 TRGCMPx Buffer Update Timing according to Trigger Output Mode Setting

<TRGxMD[2:0]> Setting	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	

12.3.8.5 PMDxTRGCR (Trigger Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit symbol	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
After reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit symbol	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as 0.
15	TRG3BE	R/W	PMDTRG3 buffer update timing 0: Sync 1: Async (The value written to PMDTRG3 is immediately reflected.) This bit enables asynchronous updating of the PMDTRG3 buffers.
14-12	TRG3MD[2:0]	R/W	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 PMDxMDCR<PMDMD> 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 <TRG3MD[2:0]>="011", PMDxTRGCMP3="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
11	TRG2BE	R/W	PMDTRG2 buffer update timing 0: Sync 1: Async (The value written to PMDTRG2 is immediately reflected.) This bit enables asynchronous updating of the PMDTRG2 buffers.
10-8	TRG2MD[2:0]	R/W	PMDTRG2 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 PMDxMDCR<PMDMD> 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 <TRG2MD[2:0]>="011", PMDxTRGCMP2="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
7	TRG1BE	R/W	PMDTRG1 buffer update timing 0: Sync 1: Async (The value written to PMDTRG1 is immediately reflected.) This bit enables asynchronous updating of the PMDTRG1 buffers.
6-4	TRG1MD[2:0]	R/W	PMDTRG1 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 PMDxMDCR<PMDMD> 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 <TRG1MD[2:0]>="011", PMDxTRGCMP1="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
3	TRG0BE	R/W	PMDTRG0 buffer update timing 0: Sync 1: Async (The value written to PMDTRG0 is immediately reflected.) This bit enables asynchronous updating of the PMDTRG0 buffers.

Bit	Bit Symbol	Type	Function
2-0	TRG0MD[2:0]	R/W	PMDTRG0 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 PMDxMDCR<PMDMD> 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 <TRG0MD[2:0]>="011", PMDxTRGCMP0="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.

12.3.8.6 PMDxTRGMD (Trigger Output Mode Setting Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
1	TRGOUT	R/W	Trigger output mode 0: Fixed trigger output 1: Variable trigger output When <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 PMDxTRGCMP0 is switched according to the <TRGSEL> setting or sector information from the Vector Engine. For details, see the table below.
0	EMGTGE	R/W	Output enable in EMG protection state 0: Disable trigger output in the protection state 1: Enable trigger output in the protection state This bit enables or disables trigger output in the EMG protection state.

Table 12-5 Trigger Output Patterns

<TRGOUT> Setting	Compare Register	<TRGSEL[2:0]> Setting	Trigger Output
<TRGOUT>=0	PMDxTRGCMP0	×	PMDTRG0
	PMDxTRGCMP1		PMDTRG1
	PMDxTRGCMP2		PMDTRG2
	PMDxTRGCMP3		PMDTRG3
<TRGOUT>=1	PMDxTRGCMP0	0	PMDTRG0
		1	PMDTRG1
		2	PMDTRG2
		3	PMDTRG3
		4	PMDTRG4
		5	PMDTRG5
	PMDxTRGCMP1	×	No trigger output
	PMDxTRGCMP2	×	No trigger output
	PMDxTRGCMP3	×	No trigger output

12.3.8.7 PMDxTRGSEL (Trigger Output Select Register)

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

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as 0.
2-0	TRGSEL[2:0]	R/W	Trigger output select 000: Output from 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 (PMDxTRGMD<TRGOUT>="1"). The selected trigger is output by a match between the PMD counter and the PMDxTRGCMP0 value. (See Table 12-5.)

13. Vector Engine (VE)

13.1 Overview

13.1.1 Features

The Vector Engine provides the following features:

1. Executes basic tasks for vector control (coordinate conversion, phase conversion and SIN/COScomputation).

   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).

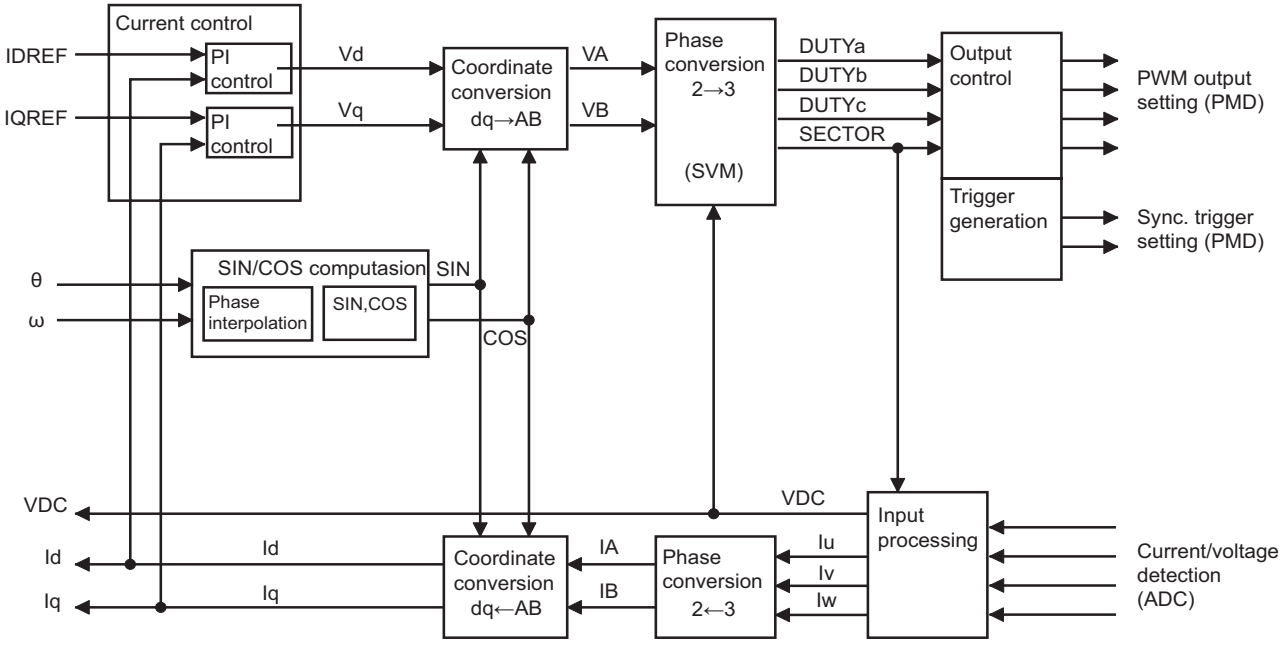


Figure 13-1 Block Diagram of Vector Control

13.1.2 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.
5. Phase information can be directly specified or computed from rotation speed by using phase interpolation.

Note 1: For using the Vector Engine, the PMD must be set to the VE mode through the mode select register (PMDxMODESEL).

Note 2: 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.

13.2 Configuration

Figure 13-2 shows the configuration of the Vector Engine.

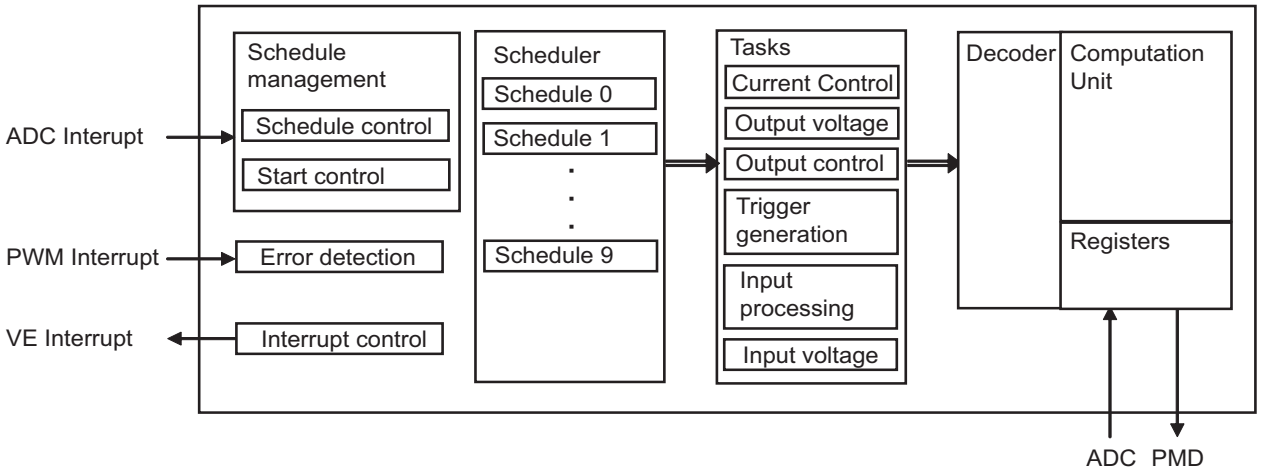


Figure 13-2 Configuration of the Vector Engine

13.2.1 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 13-3, 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 ADABPSETn<UVWISn0[1:0]>, <UVWISn1[1:0]>, <UVWISn2[1:0]>, <UVWISn3[1:0]> which are read into the Vector Engine as the Vector Engine registers VEADREG0A, VEADREG1A, VEADREG2A, VEADREG3A, VEPHNUM0A, VEPNNUM1A, 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.

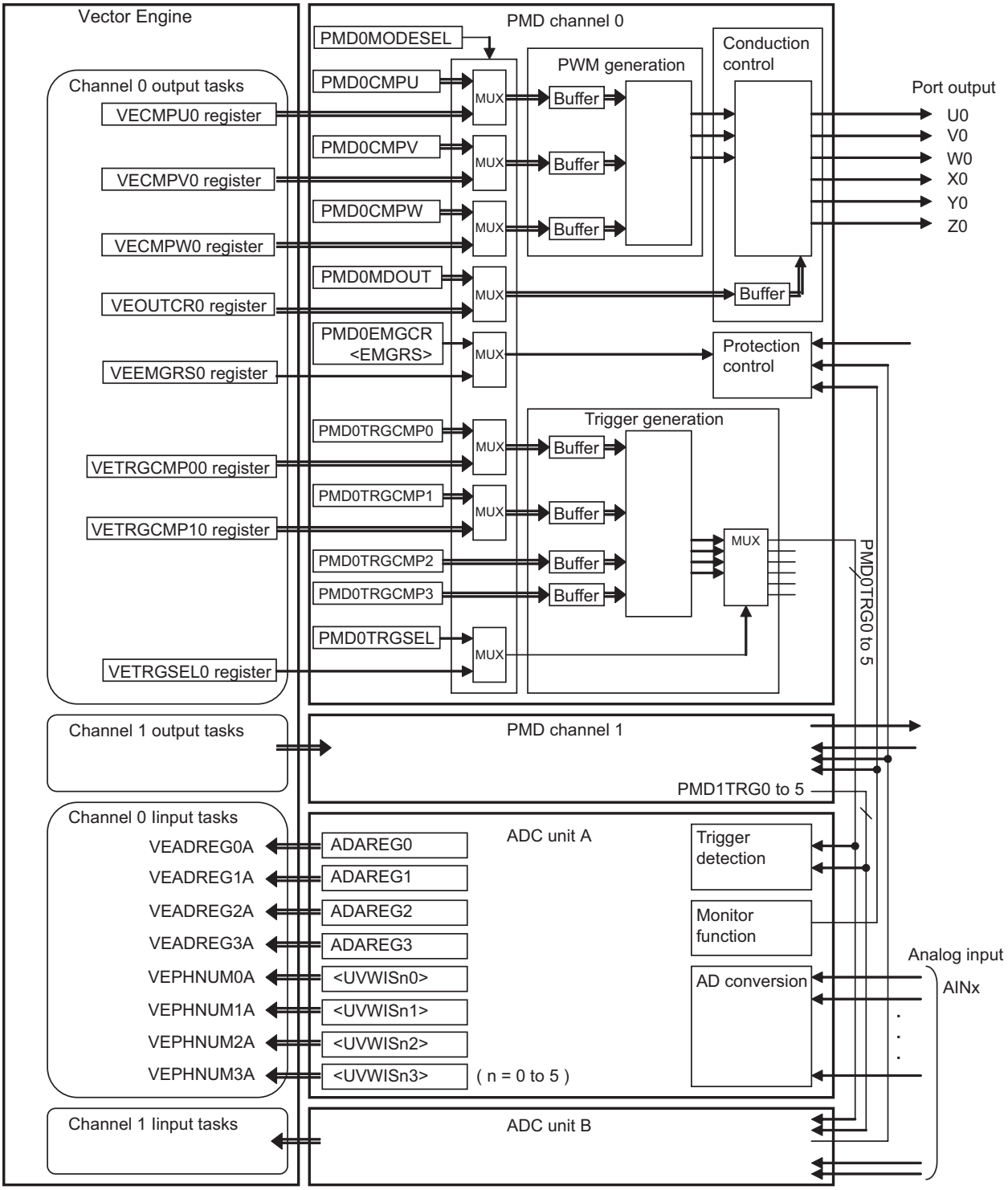


Figure 13-3 Interaction among Vector Engine, PMD and ADC

13.3 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

13.3.1 List of Registers

VE Control Registers

Register Name			Address
VE enable/disable	VEEN	R/W	0x4005_0000
CPU start trigger selection	VECPURUNTRG	W	0x4005_0004
Task selection	VETASKAPP	R/W	0x4005_0008
Operation schedule selection	VEACTSCH	R/W	0x4005_000C
Schedule repeat count	VEREPTIME	R/W	0x4005_0010
Start trigger mode	VETRGMODE	R/W	0x4005_0014
Error interrupt enable/disable	VEERRINTEN	R/W	0x4005_0018
VE forced termination	VECOMPEND	W	0x4005_001C
Error detection	VEERRDET	R	0x4005_0020
Schedule executing flag/executing task	VESCHTASKRUN	R	0x4005_0024
Reserved	-	R	0x4005_0028
Temporary 0	VETMPREG0	R/W	0x4005_002C
Temporary 1	VETMPREG1	R/W	0x4005_0030
Temporary 2	VETMPREG2	R/W	0x4005_0034
Temporary 3	VETMPREG3	R/W	0x4005_0038
Temporary 4	VETMPREG4	R/W	0x4005_003C
Temporary 5	VETMPREG5	R/W	0x4005_0040
Reserved	-	R	0x4005_01BC

Common Registers

Register Name			Address
Reserved	-	R/W	0x4005_0174
ADC conversion time (based on PWM clock)	VETADC	R/W	0x4005_0178

Channel-Specific Registers for Channel 0

Register Name			Address
Status flags	VEMCTLF0	R/W	0x4005_0044
Task control mode	VEMODE0	R/W	0x4005_0048
Flow control	VEFMODE0	R/W	0x4005_004C
PWM period rate (PWM period [s] × maximum speed (note1)×2^{16}) setting	VETPWM0	R/W	0x4005_0050
Rotation speed (speed [Hz]÷ maximum speed(note1)×2^{15}) setting	VEOMEGA0	R/W	0x4005_0054
Motor phase (motor phase [deg]/360×2^{16}) setting	VETHETA0	R/W	0x4005_0058
d-axis reference value (current[A]÷maximum current (note2)×2^{15}) setting	VEIDREF0	R/W	0x4005_005C
q-axis reference value (current[A]÷maximum current (note2)×2^{15}) setting	VEIQREF0	R/W	0x4005_0060
d-axis voltage (voltage[V]÷maximum voltage (note3)×2^{31}) setting	VEVD0	R/W	0x4005_0064
q-axis voltage (voltage[V]÷maximum voltage (note3)×2^{31}) setting	VEVQ0	R/W	0x4005_0068
Integral coefficient for PI control of d-axis	VECIDKI0	R/W	0x4005_006C
Proportional coefficient for PI control of d-axis	VECIDKP0	R/W	0x4005_0070
Integral coefficient for PI control of q-axis	VECIQKI0	R/W	0x4005_0074
Proportional coefficient for PI control of q-axis	VECIQKP0	R/W	0x4005_0078
Upper 32 bits of integral term (VDI) of d-axis voltage	VEVDIH0	R/W	0x4005_007C
Lower 32 bits of integral term (VDI) of d-axis voltage	VEVDILH0	R/W	0x4005_0080
Upper 32 bits of integral term (VQI) of q-axis voltage	VEVQIH0	R/W	0x4005_0084
Lower 32 bits of integral term (VQI) of q-axis voltage	VEVQILH0	R/W	0x4005_0088
Switching speed (for 2-phase modulation and shift PWM)	VEFPWMCHG0	R/W	0x4005_008C
PWM period (to be set identically with PMD's PWM period)	VEMDPRD0	R/W	0x4005_0090
Minimum pulse width	VEMINPLS0	R/W	0x4005_0094
Synchronizing trigger correction value	VETRGCRC0	R/W	0x4005_0098
Reserved	-	R/W	0x4005_009C
Cosine value at THETA for output conversion (Q15 data)	VECOS0	R/W	0x4005_00A0
Sine value at THETA for output conversion (Q15 data)	VESIN0	R/W	0x4005_00A4
Previous cosine value for input processing (Q15 data)	VECOSM0	R/W	0x4005_00A8

Channel-Specific Registers for Channel 0

Register Name			Address
Previous sine value for input processing (Q15 data)	VESINM0	R/W	0x4005_00AC
Sector information	VESECTOR0	R/W	0x4005_00B0
Previous sector information for input processing	VESECTORM0	R/W	0x4005_00B4
AD conversion result of a-phase zero-current (note4)	VEIAO0	R/W	0x4005_00B8
AD conversion result of b-phase zero-current (note4)	VEIBO0	R/W	0x4005_00BC
AD conversion result of c-phase zero-current (note4)	VEICO0	R/W	0x4005_00C0
AD conversion result of a-phase current (note4)	VEIAADC0	R/W	0x4005_00C4
AD conversion result of b-phase current (note4)	VEIBADC0	R/W	0x4005_00C8
AD conversion result of c-phase current (note4)	VEICADC0	R/W	0x4005_00CC
DC supply voltage (voltage[V]÷maximum voltage (note3)×2^{15})	VEVDC0	R/W	0x4005_00D0
d-axis current (current[A]÷maximum current (note2)×2^{31})	VEID0	R/W	0x4005_00D4
q-axis current (current[A]÷maximum current (note2)×2^{31})	VEIQ0	R/W	0x4005_00D8
PMD control: CMPU setting	VECMPU0	R/W	0x4005_017C
PMD control: CMPV setting	VECMPV0	R/W	0x4005_0180
PMD control: CMPW setting	VECMPW0	R/W	0x4005_0184
PMD control: Output control (MDOUT)	VEOUTCR0	R/W	0x4005_0188
PMD control: TRGCMP0 setting	VETRGCMP00	R/W	0x4005_018C
PMD control: TRGCMP1 setting	VETRGCMP10	R/W	0x4005_0190
PMD control: Trigger selection	VETRGSEL0	R/W	0x4005_0194
PMD control: EMG return	VEEMGRS0	W	0x4005_0198

Channel-Specific Registers for Channel 1

Register Name			Address
Status flags	VEMCTLF1	R/W	0x4005_00DC
Task control mode	VEMODE1	R/W	0x4005_00E0
Flow control	VEFMODE1	R/W	0x4005_00E4
PWM period rate (PWM period [s] × maximum speed (note1)×2^{16}) setting	VETPWM1	R/W	0x4005_00E8
Rotation speed (speed [Hz]÷ maximum speed(note1)×2^{15}) setting	VEOMEGA1	R/W	0x4005_00EC
Motor phase (motor phase [deg]/360×2^{16}) setting	VETHETA1	R/W	0x4005_00F0
d-axis reference value (current[A]÷maximum current (note2)×2^{15}) setting	VEIDREF1	R/W	0x4005_00F4
q-axis reference value (current[A]÷maximum current (note2)×2^{15}) setting	VEIQREF1	R/W	0x4005_00F8
d-axis voltage (voltage[V]÷maximum voltage (note3)×2^{31}) setting	VEVD1	R/W	0x4005_00FC
q-axis voltage (voltage[V]÷maximum voltage (note3)×2^{31}) setting	VEVQ1	R/W	0x4005_0100
Integral coefficient for PI control of d-axis	VECIDKI1	R/W	0x4005_0104
Proportional coefficient for PI control of d-axis	VECIDKP1	R/W	0x4005_0108
Integral coefficient for PI control of q-axis	VECIQKI1	R/W	0x4005_010C
Proportional coefficient for PI control of q-axis	VECIQKP1	R/W	0x4005_0110
Upper 32 bits of integral term (VDI) of d-axis voltage	VEVDIH1	R/W	0x4005_0114
Lower 32 bits of integral term (VDI) of d-axis voltage	VEVDILH1	R/W	0x4005_0118
Upper 32 bits of integral term (VQI) of q-axis voltage	VEVQIH1	R/W	0x4005_011C
Lower 32 bits of integral term (VQI) of q-axis voltage	VEVQILH1	R/W	0x4005_0120
Switching speed (for 2-phase modulation and shift PWM)	VEFPWMCHG1	R/W	0x4005_0124
PWM period (to be set identically with PMD's PWM period)	VEMDPRD1	R/W	0x4005_0128

Channel-Specific Registers for Channel 1

Register Name			Address
Minimum pulse width	VEMINPLS1	R/W	0x4005_012C
Synchronizing trigger correction value	VETRGCRC1	R/W	0x4005_0130
Reserved	-	R/W	0x4005_0134
Cosine value at THETA for output conversion (Q15 data)	VECOS1	R/W	0x4005_0138
Sine value at THETA for output conversion (Q15 data)	VESIN1	R/W	0x4005_014C
Previous cosine value for input processing (Q15 data)	VECOSM1	R/W	0x4005_0140
Previous sine value for input processing (Q15 data)	VESINM1	R/W	0x4005_0144
Sector information	VESECTOR1	R/W	0x4005_0148
Previous sector information for input processing	VESECTORM1	R/W	0x4005_014C
AD conversion result of a-phase zero-current (note4)	VEIAO1	R/W	0x4005_0150
AD conversion result of b-phase zero-current (note4)	VEIBO1	R/W	0x4005_0154
AD conversion result of c-phase zero-current (note4)	VEICO1	R/W	0x4005_0158
AD conversion result of a-phase current (note4)	VEIAADC1	R/W	0x4005_015C
AD conversion result of b-phase current (note4)	VEIBADC1	R/W	0x4005_0160
AD conversion result of c-phase current (note4)	VEICADC1	R/W	0x4005_0164
DC supply voltage (voltage[V]÷maximum voltage (note3)×2^{15})	VEVDC1	R/W	0x4005_0168
d-axis current (current[A]÷maximum current (note2)×2^{31})	VEID1	R/W	0x4005_016C
q-axis current (current[A]÷maximum current (note2)×2^{31})	VEIQ1	R/W	0x4005_0170
PMD control: CMPU setting	VECMPU1	R/W	0x4005_019C
PMD control: CMPV setting	VECMPV1	R/W	0x4005_01A0
PMD control: CMPW setting	VECMPW1	R/W	0x4005_01A4
PMD control: Output control (MDOUT)	VEOUTCR1	R/W	0x4005_01A8
PMD control: TRGCMP0 setting	VETRGCMP01	R/W	0x4005_01AC
PMD control: TRGCMP1 setting	VETRGCMP11	R/W	0x4005_01B0
PMD control: Trigger selection	VETRGSEL1	R/W	0x4005_01B4
PMD control: EMG return	VEEMGRS1	W	0x4005_01B8

Note 1: Maximum speed: Maximum rotation speed [Hz] that can be controlled or operated.

Note 2: Maximum current:(Phase current value [A] which corresponds to 1 LSB of AD converter)× 2^{11}

Note 3: Maximum voltage: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of AD conveter)× 2^{12}

Note 4: AD conversion results are stored in the upper 12 bits of each 16-bit register.

13.3.2 VE Control Registers

13.3.2.1 VEEN (VE enable/disable register)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	VEIDLEN	R/W	Controls whether or not the clock is supplied to the Vector Engine in IDLE mode. 0: Inactive 1: Active
0	VEEN	R/W	Disables or enables the Vector Engine. 0: Disable 1: Enable

13.3.2.2 VECPURUNTRG (CPU start trigger selection register)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	VCPURTB	W	Starts channel 1 by programming. 0: - 1: Start
0	VCPURTA	W	Starts channel 0 by programming. 0: - 1: Start

Note 1: When "1" is written to these bits, it is cleared in the next cycle. These bits always read as 0.

Note 2: The task to be performed is determined by the settings of the VEACTSCH and VETASKAPP registers.

Note 3: If a channel under executing will be restarted, it must be terminated by VECOMPEND register before a start command executed.

13.3.2.3 VETASKAPP(Task selection register)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-4	VTASKB[3:0]	R/W	Channel 1 task selection 0x0 : Output control 0x1 : Trigger generation 0x2 : Input processing 0x3 : Input phase conversion 0x4 : Input coordinate axis conversion 0x5 : Current control 0x6 : SIN/COS computation 0x7 : Output coordinate axis conversion 0x8 : Output phase conversion 0x9 -0xF : Reserved Specifies the task to be performed when channel 1 is started by programming.
3-0	VTASKA[3:0]	R/W	Channel 0 task selection 0x0 : Output control 0x1 : Trigger generation 0x2 : Input processing 0x3 : Input phase conversion 0x4 : Input coordinate axis conversion 0x5 : Current control 0x6 : SIN/COS computation 0x7 : Output coordinate axis conversion 0x8 : Output phase conversion 0x9 -0xF : Reserved Specifies the task to be performed when channel 0 is started by programming.

Note: Only those tasks that are included in schedules can be specified.

13.3.2.4 VEACTSCH (Operation schedule selection register)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-4	VACTB[3:0]	R/W	Specifies an individual task execution or a schedule for channel 1. 0x0 : Individual task execution 0x1 : Schedule 1 0x4 : Schedule 4 0x9 : Schedule 9 Other : Reserved
3-0	VACTA[3:0]	R/W	Specifies an individual task execution or a schedule for channel 0. 0x0 : Individual task execution 0x1 : Schedule 1 0x4 : Schedule 4 0x9 : Schedule 9 Other : Reserved

13.3.2.5 VEREPTIME (Schedule repeat count)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-4	VREPB[3:0]	R/W	Specifies the repeat times a schedule is to be executed in channel 1. 0: Do not execute schedule 1 to 15: Execute schedule a specified number of times
3-0	VREPA[3:0]	R/W	Specifies the repeat times a schedule is to be executed in channel 0. 0: Do not execute schedule 1 to 15: Execute schedule a specified number of times

Note: When "0" is set, no schedule is executed.

13.3.2.6 VETRGMODE (Start trigger mode)

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

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as 0.
3-2	VTRGB[1:0]	R/W	Specifies the AD conversion end interrupt that triggers input processing in channel 1. Channel 1 trigger mode 00: Ignore both INTB0(unit A) and INTB1(unit B) 01: Start by INTB0 (unit A) 10: Start by INTB1 (unit B) 11: Start when both INTB0 (unit A) and INTB1 (unit B) occur
1-0	VTRGA[1:0]	R/W	Specifies the AD conversion end interrupt that triggers input processing in channel 0. Channel 0 trigger mode 00: Ignore both INTA0(unit A) and INTA1(unit B) 01: Start by INTA0 (unit A) 10: Start by INTA1 (unit B) 11: Start when both INTA0 (unit A) and INTA1 (unit B) occur

13.3.2.7 VEERRINTEN (Error interrupt enable/disable)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	VERRENB	R/W	Enables or disables the error detection interrupt in channel 1. 0: Disable 1: Enable
0	VERRENA	R/W	Enables or disables the error detection interrupt in channel 0. 0: Disable 1: Enable

13.3.2.8 VECOMPEND (VE forced termination)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	VCENDB	W	Forcefully terminates the currently executing schedule in channel 1. 0: – 1: Terminate
0	VCENDA	W	Forcefully terminates the currently executing schedule in channel 0. 0: – 1: Terminate

Note: When "1" is written to these bits, it is cleared in the next cycle. These bits always read as "0".

13.3.2.9 VEERRDET (Error detection)

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

Bit	Bit Symbol	Type	Function
31-2	–	R	Read as 0.
1	VERRDB	R	Channel 1 error flag 0: No error detected 1: Error detected
0	VERRDA	R	Channel 0 error flag 0: No error detected 1: Error detected

Note 1: The error flags are set when a PWM interrupt is detected during execution of a schedule (excluding standby periods waiting for a start trigger).

Note 2: The error flags are cleared by a read of this register.

13.3.2.10VESCHTASKRUN (Schedule executing flag/executing task)

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

Bit	Bit Symbol	Type	Function
31-10	–	R	Read as 0.
9-6	VRTASKB[3:0]	R	Task Number currently executing in channel 1 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 to 0xF: Reserved
5	VRSCHB	R	Schedule execution status in channel 1 0: Not executing 1: Executing
4-1	VRTASKA[3:0]	R	Task Number currently executing in channel 0 0x0: Output control 0x1: Trigger generation 0x2: Input processing 0x3: Input phase conversion 0x4: Input coordinate axis conversion 0x5: Current control 0x6: SIN/COS computation 0x7: Output coordinate axis conversion 0x8: Output phase conversion 0x9 to 0xF: Reserved
0	VRSCHA	R	Schedule execution status in channel 0 0: Not executing 1: Executing

13.3.2.11VETMPREG0 (Temporary register 0)

	31	30	29	28	27	26	25	24
bit symbol	TMPREG0							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG0							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	TMPREG0[31:0]	R/W	Temporary register 0

13.3.2.12VETMPREG1 (Temporary register 1)

	31	30	29	28	27	26	25	24
bit symbol	TMPREG1							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG1							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	TMPREG1[31:0]	R/W	Temporary register 1

13.3.2.13VETMPREG2 (Temporary register 2)

	31	30	29	28	27	26	25	24
bit symbol	TMPREG2							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG2							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG2							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	TMPREG2[31:0]	R/W	Temporary register 2

13.3.2.14VETMPREG3 (Temporary register 3)

	31	30	29	28	27	26	25	24
bit symbol	TMPREG3							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG3							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG3							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	TMPREG3[31:0]	R/W	Temporary register 3

13.3.2.15VETMPREG4 (Temporary register 4)

	31	30	29	28	27	26	25	24
bit symbol	TMPREG4							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG4							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG4							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	TMPREG4[31:0]	R/W	Temporary register 4

13.3.2.16VETMPREG5 (Temporary register 5)

	31	30	29	28	27	26	25	24
bit symbol	TMPREG5							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG5							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG5							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	TMPREG5[31:0]	R/W	Temporary register 5

13.3.3 Common Registers

13.3.3.1 VETADC (Common ADC conversion time)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	TADC[15:0]	R/W	ADC conversion time (based on PWM clock) 0x0000 to 0xFFFF : (ADC conversion time[s]÷PWM counter clock frequency[s]) Note) This register is effective when the 1-shunt current detection mode is selected and PWM shift is enabled.

13.3.4 Channel-Specific Registers(x=0 to 1)

13.3.4.1 VEMODEx (Task control mode Registers)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-4	–	R/W	Always write "0".
3-2	OCRMD[1:0]	R/W	Output control 00: Output OFF 01: Output enable 10: Reserved 11: Output OFF and EMG return
1	ZIEN	R/W	Zero-current detection 0: Disable 1: Enable
0	PVIEN	R/W	Phase interpolation 0: Disable 1: Enable

13.3.4.2 VEFMODEx(Flow control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	MREGDIS	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADCSEL		-	PMDSEL	IDMODE		SPWMEN	C2PEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-10	–	R/W	Always write "0".
9	MREGDIS	R/W	Keep the previous value of SIN/COS/SECTOR 0: effective 1: no_effective In case of no_effective, VESINMx=VESINx, VECOSMx=VECOSx, VESECTORMx=VESECTORx.
8	–	R/W	Always write "0".
7-6	ADCSEL[1:0]	R/W	Selects the ADC unit to be used. 00: Unit A 01: Unit B 10: Unit A,B 11: Unit A,B By the channel of the vector engine to be used, it sets up as follows. VE: channel 0 → ADC Unit: unit A or unit A /unit B VE: channel 1 → ADC Unit: unit B or unit A/ unit B
5	–	R/W	"Always write "0".
4	PMDSEL	R/W	Selects the PMD channel. 0: Channel 0 1: Channel 1 By the channel of the vector engine to be used, it sets up as follows. VE: channel 0 → PMD Unit: channel 0 VE: channel 1 → PMD Unit: channel 1
3-2	IDMODE	R/W	Current detection mode 00: 3-shunt 01: 2-sensor 10: 1-shunt (for up count PMDTRG) 11: 1-shunt (for down count PMDTRG)
1	SPWMEN	R/W	Enables or disables PWM shift. 0: Disable 1: Enable
0	C2PEN	R/W	Selects 3-phase or 2-phase modulation. 0: 3-phase modulation 1: 2-phase modulation

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

VEFMODE0 <IDMODE[1:0]>	VEFMODE1 <IDMODE[1:0]>	PMD0TRGCR <TRG0MD[2:0]>	PMD0TRGCR <TRG1MD[2:0]>	PMD1TRGCR <TRG0MD[2:0]>	PMD1TRGCR <TRG1MD[2:0]>
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)

13.3.4.3 VETPWMx(PWM period rate control Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	TPWM[15:0]	R/W	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: 0x0000 to 0xFFFF : PWM period [s] × Max_Hz × 2^{16} (Max_Hz : Maximum rotation speed [Hz]) (It indicates a ratio between PWM frequency and maximum rotation speed.)

13.3.4.4 VEOMEGAx(Rotation speed control Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	OMEGA[15:0]	R/W	Set a rotation speed (16-bit fixed-point data: -1.0 to 1.0) as follows: 0x0000 to 0xFFFF : Rotation speed [Hz] ÷ Max_Hz × 2^{15} (Max_Hz : Maximum rotation speed [Hz])

13.3.4.5 VETHETAx(Motor phase control Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	THETA[15:0]	R/W	Set phase data (16-bit fixed-point data: 0.0 to 1.0) as follows: Formula : Phase[deg] ÷ 360 × 2^{16}

13.3.4.6 VECOSx/VESINx/VECOSMx/VESINMx

VECOSx (Cosine value at THETA for output conversion (Q15 data))

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	COS[15:0]	R/W	Cosine value based on the THETA value (16-bit fixed-point data: -1.0 to 1.0) Cosine value: 0x0000 to 0xFFFF

VESINx (Sine value at THETA for output conversion (Q15 data))

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	SIN[15:0]	R/W	Sine value based on the THETA value (16-bit fixed-point data: -1.0 to 1.0) Sine value: 0x0000 to 0xFFFF

VECOSMx (Previous cosine value for input processing (Q15 data))

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	COSM[15:0]	R/W	Previous value of the COS register Cosine value (previous value): 0x0000 to 0xFFFF

VESINMx (Previous sine value for input processing (Q15 data))

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	SINM[15:0]	R/W	Previous value of the SIN register Sine value (previous value): 0x0000 to 0xFFFF

13.3.4.7 VEIDREFx/VEIQREFx(dq Current Reference Registers)

VEIDREFx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	IDREF[15:0]	R/W	Reference value of d-axis current (16-bit fixed-point data: -1.0 to 1.0) 0x0000 to 0xFFFF(The value to be set is : d-axis current reference[A]÷Max_I×2^{15}) Max_I: (Phase current value [A] which corresponds to 1 LSB of ADC)×2^{11}

VEIQREFx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	IQREF[15:0]	R/W	Reference value of q-axis current (16-bit fixed-point data: -1.0 to 1.0) 0x0000 to 0xFFFF(The value to be set is : q-axis current reference [A]÷Max_I×2^{15}) Max_I: (Phase current value [A] which corresponds to 1 LSB of ADC)×2^{11}

13.3.4.8 VEVDx/VEVQx(d-axis/q-axis Voltage Registers)

VEVDx

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

Bit	Bit Symbol	Type	Function
31-0	VD[31:0]	R/W	d-axis voltage (32-bit fixed-point data: -1.0 to 1.0) 0x0000_0000 to 0xFFFF_FFFF(d-axis voltage÷Max_V×2^{31}) Max_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC)×2^{12}

VEVQx

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

Bit	Bit Symbol	Type	Function
31-0	VQ[31:0]	R/W	q-axis voltage (32-bit fixed-point data: -1.0 to 1.0) 0x0000_0000 to 0xFFFF_FFFF(q-axis voltage÷Max_V×2^{31}) Max_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC)×2^{12}

13.3.4.9 VECIDKIx/VECIDKPx/VEVCIQKIx/VECIQKPx(PI Control Coefficient Registers)

VECIDKIx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	CIDKI[15:0]	R/W	Integral coefficient for PI control of d-axis: 0x0000 to 0xFFFF

VECIDKPx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	CIDKP[15:0]	R/W	Proportional coefficient for PI control of d-axis: 0x0000 to 0xFFFF

VEVCIQKIx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	CIQKI[15:0]	R/W	Integral coefficient for PI control of q-axis: 0x0000 to 0xFFFF

VECIQKPx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	CIQKP[15:0]	R/W	Proportional coefficient for PI control of q-axis: 0x0000 to 0xFFFF

13.3.4.10VEVDIHx/VEVDILHx/VEVQIHx/VEVQILHx(PI Control Integral Term Registers)

VEVDIHx

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

Bit	Bit Symbol	Type	Function
31-0	VDIH[31:0]	R/W	Upper 32 bits of the integral term (VDI) for PI control of d-axis

VEVDILHx

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

Bit	Bit Symbol	Type	Function
31-16	VDILH[15:0]	R/W	Bit 31 to 16 of the integral term (VDI) for PI control of d-axis
15-0	–	R	Read as 0.

Note: 64-bit fixed-point data with 63 fractional bits (-1.0 to 1.0)

VEVQIHx

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

Bit	Bit Symbol	Type	Function
31-0	VQIH[31:0]	R/W	Upper 32 bits of the integral term (VQI) for PI control of q-axis

VEVQILHx

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

Bit	Bit Symbol	Type	Function
31-16	VQILH[15:0]	R/W	Bit 31 to 16 of the integral term (VQI) for PI control of q-axis
15-0	–	R	Read as 0.

Note: 64-bit fixed-point data with 63 fractional bits (-1.0 to 1.0)

13.3.4.11VEMCTLFx(Status flags Register)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7-6	–	R/W	Always write "0".
5	PLSLFM	R/W	Previous value of <PLSLF>
4	PLSLF	R/W	Minimum disparity of pulse width Minimum disparity of pulse width ≥ VEMINPLSx<MINPLS> case="0" Minimum disparity of pulse width < VEMINPLSx<MINPLS> case ="1"
3	–	R/W	Always write "0".
2	LVTF	R/W	Supply voltage lower flag VEVDCx<VDC> ≥ 0x0100 (1/128) case="0" VEVDCx<VDC> < 0x0100 (1/128) case="1"
1	LAVFM	R/W	Previous <LAVF> value
0	LAVF	R/W	Low-speed flag 0: High-speed 1: Low-speed VEOMEGAx<OMEGA> ≥ VEFPWMCHGx<FPWMCHG> case="0" VEOMEGAx<OMEGA> < VEFPWMCHGx<FPWMCHG> case="1"

13.3.4.12VEFPWMCHGx(Switching speed (for 2-phase modulation and shift PWM))

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	FPWMCHG[15:0]	R/W	Rortation speed when PWM shift is enabled. The value to be set is: Rortation speed [Hz] ÷ Max_Hz × 2^{15} (Max_Hz : Maximum rotation speed [Hz])

13.3.4.13VEMDPRDx(PWM period control Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	VMDPRD[15:0]	R/W	PWM period Set the value of the PMD's PMDxMDPRD register.

13.3.4.14VEMINPLSx(Minimum pulse width)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	MINPLS[15:0]	R/W	Set the minimum disparity of pulse width among the duty of VECMPUx, VECMPVx, VECMPWx. The value to be set is : Disparity of pulse width[s] ÷ PWMcounter clock period[s]

13.3.4.15VESECTORx/VESECTORMx(Sector information Register)

VESECTORx

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

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as "0".
3-0	SECTOR[3:0]	R/W	Sector information Value : 0x0 to 0xF Indicates the rotation position at the time of output by 12 sectors each having 30 degrees.

VESECTORMx

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

Bit	Bit Symbol	Type	Function
31-4	–	R	Read as "0".
3-0	SECTORM[3:0]	R/W	revious sector information. Value : 0x0 - 0xF Used in input processing.

13.3.4.16VEIAOx/VEIBOx/VEICOx(Zero-Current Registers)

VEIAOx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	IAO[15:0]	R/W	AD conversion result of a-phase at zero-current. (Stores the AD conversion result of a-phase current when the motor is at stop.)

VEIBOx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	IBO[15:0]	R/W	AD conversion result of b-phase at zero-current. (Stores the AD conversion result of b-phase current when the motor is at stop.)

VEICOx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	ICO[15:0]	R/W	AD conversion result of c-phase at zero-current. (Stores the AD conversion result of c-phase current when the motor is at stop.)

Note 1: When the zero-current detection is enabled, AD conversion results are automatically stored to these registers.

Note 2: AD conversion results are stored in the 15-4 bits, with the 3-0 bits always "0".

13.3.4.17VEIAADCx/VEIBADCx/VEICADCx(Current ADC Result Registers)

VEIAADCx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	IAADC[15:0]	R/W	Stores the AD conversion result of a-phase current: 0x0000 to 0xFFFF

VEIBADCx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	IBADC[15:0]	R/W	Stores the AD conversion result of b-phase current: 0x0000 to 0xFFFF

VEICADCx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	ICADC[15:0]	R/W	Stores the AD conversion result of c-phase current: 0x0000 to 0xFFFF

Note: AD conversion results are stored in the 15-4 bits, with the 3-0 bits always "0".

13.3.4.18VEVDCx(Supply Voltage Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	VDC[15:0]	R/W	Supply voltage (16-bit fixed-point data: 0 to 1.0) Value : 0x0000 to 0xFFFF The actual voltage value is: VDC value×Max_V value÷2^{15} Max_V : (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC)×2^{12}

13.3.4.19VEIDx/VEIQx(d-axis/q-axis Current Registers)

VEIDx

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

Bit	Bit Symbol	Type	Function
31-0	ID[31:0]	R/W	d-axis current (32-bit fixed-point data: -1.0 to 1.0) d-axis current: 0x0000_0000 to 0xFFFF_FFFF The actual current value is: ID value×Max_I value÷2^{31} Max_I : (Phase current value [A] which corresponds to 1 LSB of ADC)×2^{11}

VEIQx

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

Bit	Bit Symbol	Type	Function
31-0	IQ[31:0]	R/W	q-axis current (32-bit fixed-point data: -1.0 to 1.0) q-axis current: 0x0000_0000 to 0xFFFF_FFFF The actual current value is: IQ value×Max_I value÷2^{31} Max_I : (Phase current value [A] which corresponds to 1 LSB of ADC)×2^{11}

13.3.4.20VECMPUx / VECMPVx/ VECMPWx(PWM Duty Register)

VECMPUx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	VCMPU[15:0]	R/W	PWM setting of U-phase PWM pulse width of U-phase: 0x0000 to 0xFFFF

VECMPVx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	VCMPV[15:0]	R/W	PWM setting of V-phase PWM pulse width of V-phase: 0x0000 to 0xFFFF

VECMPWx

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	VCMPW[15:0]	R/W	PWM setting of W-phase PWM pulse width of W-phase: 0x0000 to 0xFFFF

13.3.4.21VEOUTCRx(6-Phase Output Control Register)

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

Bit	Bit Symbol	Type	Function
31-9	–	R	Read as "0".
8	WPWM	R/W	PWM of W-phase 0: ON/OFF output 1: PWM output
7	VPWM	R/W	PWM of V-phase 0: ON/OFF output 1: PWM output
6	UPWM	R/W	PWM of U-phase 0: ON/OFF output 1: PWM output
5-4	WOC[1:0]	R/W	Output control of W-phase 00: WO OFF, ZO OFF 01: WO ON, ZO OFF 10: WO OFF, ZO ON 11: WO ON, ZO ON (Note) WO and ZO are both ON when <WPWM>=1.
3-2	VOC[1:0]	R/W	Output control of V-phase 00: VO OFF, YO OFF 01: VO ON, YO OFF 10: VO OFF, YO ON 11: VO ON, YO ON (Note) VO and YO are both ON when <VPWM>=1.
1-0	UOC[1:0]	R/W	Output control of U-phase 00: UO OFF, XO OFF 01: UO ON, XO OFF 10: UO OFF, XO ON 11: UO ON, XO ON (Note) UO and XO are both ON when <UPWM>=1.

Output control of U,V and W-phase of PMD is shown below. (The table shows only those combinations that are used in the VE.)

<UPWM>, <UOC> PMD setting: Output control of U-phase (UO,XO)

Setting		Output	
<UPWM>	<UOC>	UO	XO
0	00	OFF output	OFF output
1	00	PWMU inverted output	PWMU output
1	11	PWMU output	PWMU inverted output

<VPWM>,<VOC> PMD setting: Output control of V-phase (VO,YO)

Setting		Output	
<VPWM>	<VOC>	VO	YO
0	00	OFF output	OFF output
1	00	PWMV inverted output	PWMV output
1	11	PWMV output	PWMV inverted output

<WPWM>,<WOC> PMD setting: Output control of W-phase (WO,ZO)

Setting		Output	
<WPWM>	<WOC>	WO	ZO
0	00	OFF output	OFF output
1	00	PWMW inverted output	PWMW output
1	11	PWMW output	PWMW inverted output

13.3.4.22VETRGCRCx(Synchronizing trigger correction value Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	TRGCRC[15:0]	R/W	Used to correct the synchronizing trigger timing. The value to be set is: Correction time[s] ÷ PWM counter clock frequency[s]

13.3.4.23VETRGCMP0x/VETRGCMP1x(Trigger timing setting Register)

VETRGCMP0x

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	VTRGCMP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	VTRGCMP0							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	VTRGCMP0[15:0]	R/W	PMD setting: Specifies the trigger timing for sampling ADC in synchronization with PMD. 0x0000: Prohibited 0x0001 to (<MDPRD[15:0]> value -1): Trigger timing <MDPRD[15:0]> value to 0xFFFF: Prohibited

VETRGCMP1x

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	VTRGCMP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	VTRGCMP1							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as 0.
15-0	VTRGCMP1[15:0]	R/W	PMD setting: Specifies the trigger timing for sampling ADC in synchronization with PMD. 0x0000: Prohibited 0x0001 to (<MDPRD[15:0]> value -1): Trigger timing <MDPRD[15:0]> value to 0xFFFF: Prohibited

Note 1: These registers are effective when one of the following PMD trigger modes is selected: count-down match, count-up match, count-up/-down match.

Note 2: These registers are ineffective when the PMD trigger output mode is set to trigger select output (PMDxTRGMD<TRGOUT>=1).

13.3.4.24VETRGSELx(Synchronizing trigger selection Register)

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

Bit	Bit Symbol	Type	Function
31-3	–	R	Read as 0.
2-0	VTRGSEL[2:0]	R/W	PMD setting: Specifies the synchronizing trigger number to be output at the timing specified in the <VTRGCMP0[15:0]>. 0 to 5: Output trigger number 6 to 7: Prohibited Note) These registers are effective when the PMD trigger output mode is set to trigger select output (PMDxTRGMD<TRGOUT>= 1).

13.3.4.25VEEMGRSx(EMG return control Register)

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

Bit	Bit Symbol	Type	Function
31-1	–	R	Read as "0".
0	EMGRS	R/W	PMD setting : EMG return command for returning from the EMG state 0: Nop 1: EMG return command

13.4 Description of Operations

13.4.1 Schedule Management

Figure 13-4 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.

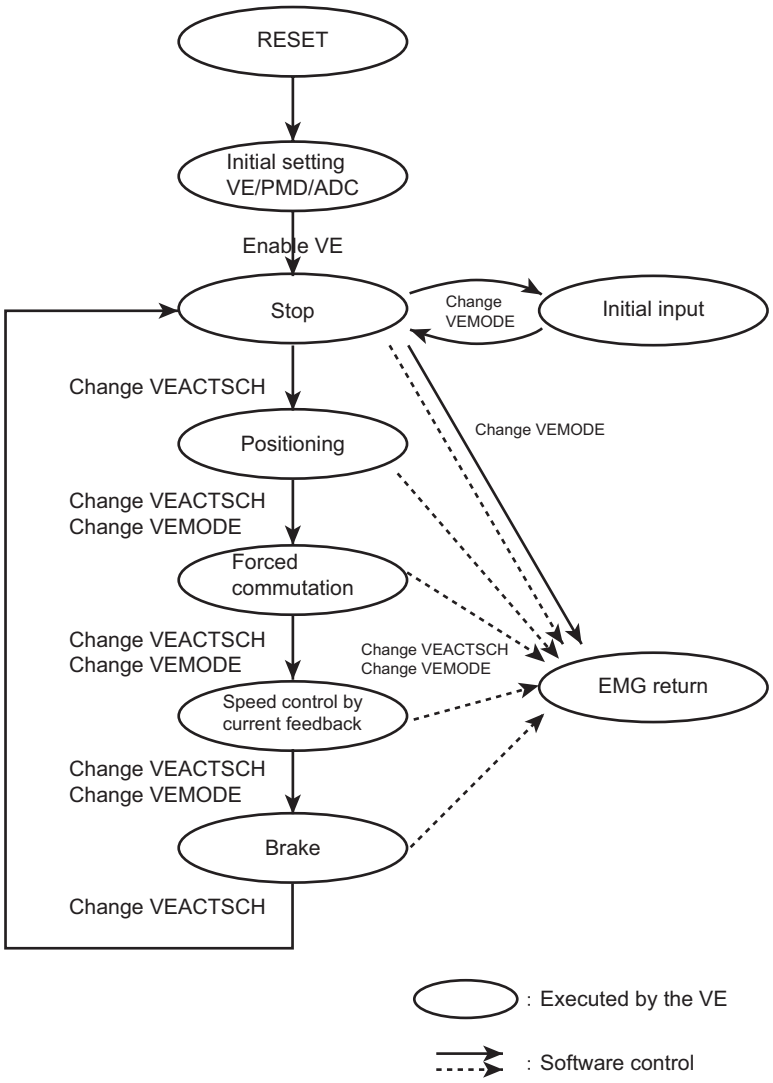


Figure 13-4 Example of Motor Control Flow

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.

13.4.1.1 Schedule Control

The VEACTSCH register is used to select the schedule to be executed.

A schedule is comprised of an output schedule handling output-related tasks and an input schedule handling input-related tasks. Table 13-1 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 13-2).

Table 13-1 Tasks To Be Executed in Each Schedule

Schedule Selection VEACTSCH	Output Schedule						Input Schedule		
	Current control	SIN/COS computation	Output coordinate axis conversion	Output phase conversion	Output control	Trigger generation	Input processing	Input phase conversion	Input coordinate axis conversion
0 : Individual execution	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)
1 : Schedule 1	o	o (Note2)	o	o	o (Note3)	o	o(Note4)	o	o
4 : Schedule 4	-	o (Note2)	o	o	o(Note3)	o	o (Note4)	o	o
9 : Schedule 9	-	-	-	-	o (Note3)	o	o(Note4)	-	-

Note 1: Each task is executed only when it is specified.
Note 2: Phase interpolation.
Note 3: Output OFF: <EMGRS>
Note 4: Task operation to be switched by zero-current detection.

Table 13-2 Typical Setting Example

Register Setting / Motor Control Flow	Schedule selection VEACTSCH <VACTn[3:0]>	Task specification VETASKAPP <VTASKn[3:0]>	Phase interpolation VEMODE <PVIEN>	Output control VEMODE <OCRMD[1:0]>	Zero-current detection VEMODE <ZIEN>
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.

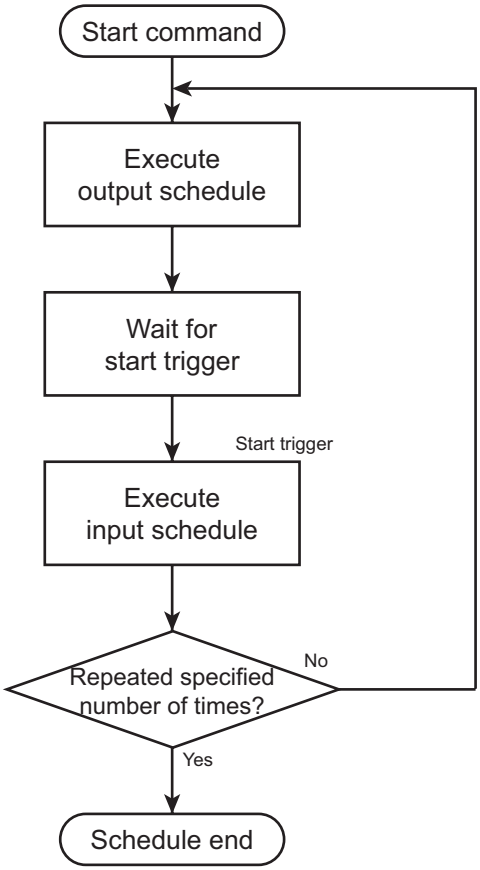


Figure 13-5 Schedule Execution Flow

13.4.1.2 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:
  1. By the VECPURUNTRG command. In this case, the task specified in the VETASKAPP register is executed.
  2. On a repeat start (when VEREPTIME ≥ 2) after the corresponding input schedule is completed.

- An input schedule is started:
  1. By a start trigger (selected in the VETRGMODE register) after the corresponding output schedule is completed.
  2. By the VECPURUNTRG command. In this case, the task specified in the VETASKAPP register is executed.

13.4.2 Summary of Tasks

Table 13-3 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 13-3 List of Tasks

Task		Task Description	Task Number
Output schedule	Current control	Controls dq currents	5
	SIN/COS computation	Performs sine/cosine computation and-phase interpolation.	6
	Output coordinate axis conversion	Converts dq coordinates to αβ coordinates.	7
	Output phase conversion	Converts 2-phase to 3-phase.	8
	Output control	Converts data to PMD setting format. Switches PWM shift.	0
	Trigger generation	Generates synchronization trigger timing.	1
Input schedule	Input processing	Captures AD conversion results and converts them into fixed-point format.	2
	Input phase conversion	Converts 3-phase to 2-phase.	3
	Input coordinate axis conversion	Converts αβ coordinates to dq coordinates.	4

13.4.2.1 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.

1. PI control of d-axis current

<Equations>

$\Delta ID = VEIDREFx - <ID[31:0]>$: Difference between current reference value and current feedback

$VDIx = VECIDKIx \times \Delta ID + VDIx$: Integral term computation

$VEVDx = VECIDKPx \times \Delta ID + VDIx$: Voltage calculation using proportional term

	Register Name	Function	
Input	VEIDx	d-axis current	32-bit fixed-point data (31 fractional bits)
	VEIDREFx	eference value of d-axis current	16-bit fixed-point data (15 fractional bits)
	VECIDKPx	Proportional coeffi-cient	16-bit data
	VECIDKIx	Integral coefficient	16-bit data
Output	VEVDx	d-axis voltage	32-bit fixed-point data (31 fractional bits)
Internal	VDIx	Integral term of d-axis voltage	64-bit fixed-point data (63 fractional bits)

2. PI control of q-axis current

<Equations>

$\Delta IQ = VEIQREFx - <IQ[31:0]>$: Difference between current reference value and current feedback

$VQIx = VECIQKIx \times \Delta IQ + VQIx$: Integral term computation

$VEVQx = VECIQKPx \times \Delta IQ + VQIx$: Voltage calculation using proportional term

	Register Name	Function	
Input	VEIQx	q-axis current	32-bit fixed-point data (31 fractional bits)
	VEIQREFx	Reference value of q-axis current	16-bit fixed-point data (15 fractional bits)
	VECIQKPx	Proportional coeffi-cient	16-bit data
	VECIQKIx	Integral coefficient	16-bit data
Output	VEVQx	q-axis voltage	32-bit fixed-point data (31 fractional bits)
Internal	VQIx	Integral term of q-axis voltage	64-bit fixed-point data (63 fractional bits)

13.4.2.2 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.

1. Phase interpolation

<Equations>

VETHETAx = VEOMEGAx × VETPWMx + VETHETAx : Integration of rotation speed. Only when phase interpolation is enabled.

	Register Name	Function	
Input	VETHETAx	Phase θ	16-bit fixed-point data (0.0 to1.0, 16 fractional bits)
	VEOMEGAx	Rotation speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VETPWMx	PWM period rate	16-bit fixed-point data (0.0 to1.0, 16 fractional bits)
	VEMODEx	Phase interpolation enable	Mode settings
Output	VETHETAx	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)

2. SIN/COS computation

<Equations>

VESINMx = VESINx : Saves previous value (for input processing).

VECOSMx = VECOSx : Saves previous value (for input processing).

VESINx = sin (VETHETAx × π) : SIN/COS computation

VECOSx = sin ((VETHETAx + 1/4) × π) : SIN/COS computation

	レジスタ名	Function	
Input	VETHETAx	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
OUTPUT	VESINx	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOSx	Cosine value at θ	
	VESINMx	Previous sine value	
	VECOSMx	Previous cosine value	

13.4.2.3 Output Voltage Conversion (Coordinate axis Conversion/Phase Conversion)

Output voltage conversion is comprised of dq-to-αβ coordinate axis conversion and 2-phase-to-3-phase conversion.

The dq-to-αβ coordinate axis conversion calculates Vα,Vβ from Vd, Vq in SIN and COS.

The 2-phase-to-3-phase conversion performs segmentation by using Vα and Vβ and performs space vector conversion to calculate Va, Vb and Vc.

For the 2-phase-to-3-phase conversion, either 2-phase modulation or 3-phase modulation can be selected.

1. dq-to-αβ coordinate conversion

<Equations>

VETMPREG3 = VECOSx × VEVDx − VESINx × VEVQx : Calculates Vα.

VETMPREG4 = VESINx × VEVDx + VECOSx × VEVQx : Calculates Vβ.

	Register Name	Function	
INPUT	VEVDx	d-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEVQx	q-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VESINx	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOSx	Cosine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
OUTPUT	VETMPREG3	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)

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

a. Segmentation

<Equations>

VESECTORMx = VESECTORx : Saves previous sector.

if (Vα ≥ 0 & Vβ ≥ 0)
  if (|Vα| ≥ |Vβ| ÷ SQR(3)
    if (|Vα| ÷ SQR(3) ≥ |Vβ|) <SECTOR[3:0]>=0
    else <SECTOR[3:0]>=1
  else <SECTOR[3:0]>=2
else if (Vα < 0 & Vβ ≥ 0)
  if (|Vα| < |Vβ| ÷SQR(3)) <SECTOR[3:0]>=3
  else if (|Vα| ÷SQR(3) < |Vβ|) <SECTOR[3:0]>=4
    else <SECTOR[3:0]>=5
else if (Vα < 0 & Vβ < 0)
  if (|Vα| ≥ |Vβ| ÷SQR(3))
    if (|Vα| ÷SQR(3) ≥ |Vβ|) <SECTOR[3:0]>=6
    else <SECTOR[3:0]>=7
  else <SECTOR[3:0]>=8
else if (Vα ≥ 0 & Vβ < 0)
  if (|Vα| < |Vβ| ÷SQR(3)) <SECTOR[3:0]>=9
  else if (|Vα| ÷SQR(3) < |Vβ|) <SECTOR[3:0]>=10
    else <SECTOR[3:0]>=11

	Register Name	Function	
Input	VETMPREG3	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
Output	VESECTORx	Sector	4-bit data
	VESECTORMx	Previous sector	4-bit data

b. 3-phase voltage calculation (when 3-phase modulation is selected and <SECTOR[3:0]>=0)

<Equations>

$t1 = (\sqrt{3})/(VEVDC) \times ((\sqrt{3})/2 \times V\alpha - 1/2 \times V\beta)$: Calculates V1 period.

$t2 = (\sqrt{3})/(VEVDC) \times (V\beta)$: Calculates V2 period.

t3 = 1 – t1 – t2 : Calculates V0+V7 period.

VETMPREG0 = t1 + t2 + t3 ÷ 2 : Calculates Va.

VETMPREG1 = t2 + t3 ÷ 2 : Calculates Vb.

VETMPREG2 = t3 ÷ 2 : Calculates Vc.

	Register Name	Function	
Input	VETMPREG3	α-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEVDCx	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VESECTORx	Sector	4-bit data
	VEFMODEx	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)

13.4.2.4 Output Control

The output control unit converts 3-phase voltage values into PWM setting format (VECMPUx, VECMPVx and VECMPWx), and sets the VEOUTCRx 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	Function	
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)
	VEMDPRDx	PWM period	16-bit data (PMD PWM period)
	VESECTORx	Sector	4-bit data
	VEOMEGAx	Rotation speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEFPWMCHGx	PWM shift switching reference	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEMODEx	Output control operation	Mode settings
	VEFMODEx	PMD channel/ shift enable/ modulation mode/ detection mode/	Mode settings
Output	VECMPUx	PMD U-phase PMW setting	16- bit data (0 to MDPRD value)
	VECMPVx	PMD V-phase PWM setting	16- bit data (0 to MDPRD value)
	VECMPWx	PMD W-phase PWM setting	16- bit data (0 to MDPRD value)
	VEOUTCRx	PMD output control setting	9-bit setting
	VEEMGRSx	PMD EMG return	1-bit setting
	VEMCTLFx	Shift switching flag	Status

13.4.2.5 Trigger Generation

The trigger generation unit calculates the trigger timing from the PWM setting values (VECMPUx, VECMPV and VECMPW) as appropriate to the current detection method, and sets the VETRGCMP0x and VETRGCMP1x registers.

	Register Name	Function	
Input	VECMPUx	PMD U-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPVx	PMD V-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPWx	PMD W-phase PWM setting	16-bit data (0 to MDPRD value)
	VEMDPRDx	PWM period setting	16-bit data (PMD PWM period)
	VETADC	AD conversion time	16-bit data (0 to MDPRD value)
	VETRGCRCx	Trigger correction value	16-bit data (0 to MDPRD value)
	VESECTORx	Sector	4-bit data
	VEMODEx	Output control operation	Mode settings
	VEFMODEx	PMD channel/ shift enable/ modulation mode/ detection mode	Mode settings
	VEMCTLFx	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)
	VETRGSELx	PMD trigger selection	3-bit data

13.4.2.6 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	Function	
Input	VEADREG0A	ADC unit A conversion result 0	16-bit data (The upper 12 bits are used.)
	VEADREG1A	ADC unit A conversion result 1	
	VEADREG2A	ADC unit A conversion result 2	
	VEADREG3A	ADC unit A conversion result 3	
	VEADREG0B	ADC unit B conversion result 0	16-bit data (The upper 12 bits are used.)
	VEADREG1B	ADC unit B conversion result 1	
	VEADREG2B	ADC unit B conversion result 2	
	VEADREG3B	ADC unit B conversion result 3	
	VEPHNUM0A	ADREG0A detected phase information	2-bit data
	VEPHNUM1A	ADREG1A detected phase information	
	VEPHNUM2A	ADREG2A detected phase information	
	VEPHNUM3A	ADREG3A detected phase information	
	VEPHNUM0B	ADREG0B detected phase information	2-bit data
	VEPHNUM1B	ADREG1B detected phase information	
	VEPHNUM2B	ADREG2B detected phase information	
	VEPHNUM3B	ADREG3B detected phase information	
	VESECTORMx	Sector information	4-bit data
	VEMODEx	Zero-current detection	Mode settings
	VEFMODEx	PMD channel / current detection mode / ADC unit /shift enable	Mode settings
	VEMCTLFx	Shift switching flag	Status
Output	VEVDCx	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	
	VETMPREG2	c-phase current	
Internal	VEIAOx	a-phase zero-current conversion result	16-bit data (The upper 12 bits are used.)
	VEIBOx	b-phase zero-current conversion result	
	VEICOx	c-phase zero-current conversion result	
	VEIAADCx	a-phase current conversion result	16-bit data (The upper 12 bits are used.)
	VEIBADCx	b-phase current conversion result	
	VEICADCx	c-phase current conversion result	

13.4.2.7 Input Current Conversion (Phase Conversion/Coordinate axis Conversion)

Input current conversion is comprised of 3-phase-to-2-phase conversion and αβ-to-dq coordinate axis conversion.

The 3-phase-to-2-phase conversion calculates Iα and Iβ from la, lb and lc.

The αβ-to-dq coordinate axis conversion calculates ld and lq from Id and Iq from Iα, Iβ, VESINM and VECOSM.

1. 3-phase-to-2-phase conversion

<Equations>

VETMPREG3 = VETMPREG0 : Calculates Iα.

VETMPREG4 = 1 ÷ SQR(3) × VETMPREG1 − 1 ÷ SQR(3) × VETMPREG2 : Calculates Iβ

	Register Name	Function	
Input	VETMPREG0	a-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
Output	VETMPREG3	α-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)

2. αβ-to-dq coordinate conversion

<Equations>

VEIDx = VECOSMx × VETMPREG3 + VESINMx × VETMPREG4 : Calculates Id.

VEIQx = − VESINMx × VETMPREG3 + VECOSMx × VETMPREG4 : Calculates Iq.

	Register Name	Function	
Input	VETMPREG3	α-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VETMPREG4	β-axis current	
	VESINMx	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOSMx	Cosine value at θ	
Output	VEIDx	d-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEIQx	q-axis current	

13.5 Combinations of VE Channel, ADC Unit and PMD Channel

By the use channel of a vector engine, the combination of PMD and ADC which can be used has restriction.

The combination used also by current detection selection and use ADC unit selection changes.

Table 13-4 Combination of VE and PMD

Vector Enginet	PMD
Channel 0	Channel 0
Channel 1	Channel 1

Table 13-5 Combination of VE and ADC

Vector Engine			ADC Unit A				ADC Unit B			
Channel	VEFMODE (Note2)		ADREG0	ADREG1	ADREG2	ADREG3	ADREG0	ADREG1	ADREG2	ADREG3
	Current ditection <IDMODE[1:0]>	ADC Selection <ADCSEL[1:0]>								
0	0x	00	Current data 1	Current data 2	Note 1	VDC data	-	-	-	-
		1x	Current data 1	-	Note 1	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 1	VDC data
		1x	-	Current data 2	-	-	-	Current data 1	Note 1	VDC data
	1x	01	-	-	-	-	Current data 1	Current data 2	-	VDC data

Note 1: Specifying the phase information to the register is necessary. However the AD conversion result of its register is not used for calculation.

Note 2: Please do not use the combination of VE and ADC which is not allowed in the table.

14. Encoder Input Circuit (ENC)

14.1 Outline

The encoder input circuit supports four operation modes including encoder mode, sensor mode (two types) and timer mode. And the functions are as follows:

- Supports incremental encoders and Hall sensor ICs. (signals of Hall sensor IC can be input directly)
- 24-bit general-purpose timer mode
- Multiply-by-4 (multiply-by-6) circuit
- Rotational direction detection circuit
- 24-bit counter
- Comparator enable/disable
- Interrupt request output:1
- Digital noise filters for input signals

14.2 Differences between channels

The TMPM376FDDFG/FDFG has a two-channel incremental encoder interface (ENC0 and ENC1), which can obtain the absolute position of the motor, based on input signals from the incremental encoder.

These channels operate identical except the differences in below.

Table 14-1 Differences between channels

Channel	Input pin			Encoder input interrupt
	A-phase	B-phase	Z-phase	
Channel0	PD0 / ENCA0	PD1 / ENCB0	PD2 / ENCZ0	INTENC0
Channel1	PF2 / ENCA1	PF3 / ENCB1	PF4 / ENCZ1	INTENC1

14.3 Block Diagram

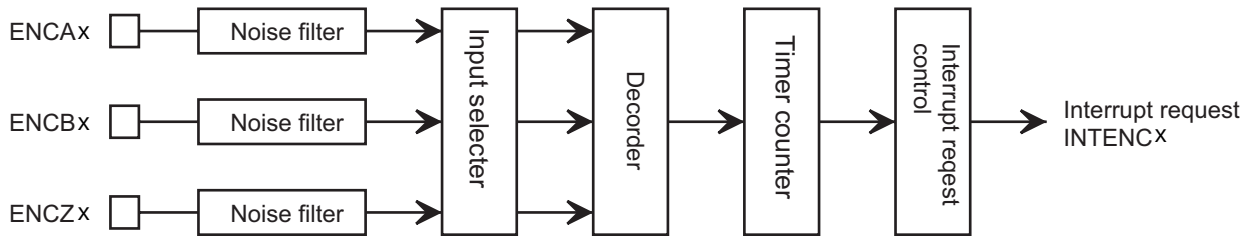


Figure 14-1 Block diagram of encoder input circuit

14.4 Registers

14.4.1 List of Registers

The following is control registers and addresses of encoder input circuit.

Channel x	Base Address
Channel0	0x4001_0400
Channel1	0x4001_0500

Register name (x=0,1)		Address(Base+)
Encoder Input Control Register	ENxTNCR	0x0000
Encoder Counter Reload Register	ENxRELOAD	0x0004
Encoder Compare Register	ENxINT	0x0008
Encoder Counter	ENxCNT	0x000C

14.4.2 ENxTNCR(Encoder Input Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	MODE		P3EN
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ZEN	ENRUN	NR		INTEN	ENDEV		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-19	–	R	Read as "0".
18-17	MODE[1:0]	R/W	Encoder input mode setting 00:Encoder mode 01:Sensor mode (event count)) 10:Sensor mode (timer count)) 11:Timer mode
16	P3EN	R/W	2-phase / 3-phase input selection (sensor mode) (Note 1) 0:2-phase input 1:3-phase input Sets the number of input signals.
15	CMP	R	Compare flag 0:- 1:Compare (Clear by RD) If comparing is executed, <CMP> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect.
14	REVERR	R	Reverse error flag (Sensor mode (at timer count)) (Note 2) 0:- 1:Error (Clear by RD) In sensor mode (at timer count), when a reverse error occurs, <REVERR> is set to "1". Flag is cleared by reading the values. When <ENRUN> = "0" is set, always "0" is set. Writing to this bit is no effect. In the encoder mode, sensor mode (event count) and timer mode, this bit has no meaning.
13	UD	R	Rotation direction 0:CCW (A-phase has the 90-degree phase lead to B-phase using incremental encoder) 1:CW (A-phase has the 90-degree phase lag to B-phase using incremental encoder) <UD> is set to "0", when <ENRUN> = "0".
12	ZDET	R	Z-Detected 0:Not detected 1:Z-phase detected <ZDET> is set to 1 on the first edge of Z input signal (ENCZ) after <ENRUN> is written from 0 to 1. This occurs on a rising edge of the signal Z during CW rotation or on a falling edge of Z during CCW rotation. <ZDET> is set to "0" when <ENRUN> = "0". <ZEN> has no influence on the value of <ZDET>. <ZDET> is set to "0" in the sensor event count and the sensor timer count modes. In the sensor mode (event count) and sensor mode (timer count), this bit is always set to "0".
11	SFTCAP	W	Executes software capture (timer mode/sensor mode (at timer count)) 0:- 1:Software capture If <SFTCAP> is set to 1, the value of the encoder counter is captured into the ENxCNT register. Writing "0" to <SFTCAP> has no effect. Reading <SFTCAP> always returns to "0". In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; writing "1" to this bit is ignored.
10	ENCLR	W	Encoder pulse counter clear 0:- 1:Clear Writing a 1 to <ENCLR> clears the encoder counter to "0". Once cleared, the encoder counter restarts counting from 0. Writing "0" to <ENCLR> has no effect. Reading <ENCLR> always returns to "0".
9	ZESEL	R/W	Edge selection of ENCZ (timer mode) 0:Rising edge 1:Falling edge In timer mode, this bit selects inputs edge of ENCZ used as external trigger. In the other mode, this bit has no meaning.
8	CMPEN	R/W	Compare enable 0:Disable 1:Enable When "1" is set to <CMPEN>, this bit compares counter values of encoder counter with register value of ENINT. When "0" is set to <CMPEN> , this compare is disabled.

Bit	Bit Symbol	Type	Function
7	ZEN	R/W	Z-phase enable (Encoder mode/timer mode) 0:Disable 1:Enable In the other mode, this bit has no meaning <Encoder mode> Clear setting of encoder counter using ENCZ input: When <ZEN> = "1" is set, if a rising edge of ENCZ is detected during rotating clockwise, the encoder counter is cleared to "0". If a falling edge of ENCZ is detected during rotating counter- clockwise, the encoder counter is cleared to "0". If the edges of ENCLK (multiply by 4 clock derived from the decoded A and B signals) and the edge of ENCZ coincide, the encoder counter is cleared to "0" without incrementing or decrementing (i.e., the clear takes precedence). <Timer mode> Sets ENCZ input to use as an external trigger.: When <ZEN> = 1, the value of the encoder counter is captured into the EN0INT register and cleared to "0" on the edge of ENCZ selected by <ZESEL>.
6	ENRUN	R/W	Encoder operation enable 0:Disable 1:Enable Setting <ENRUN> to 1 and clearing <ZDET> to "0" enables the encoder operation. Clearing <ENRUN> to "0" disables the encoder operation. There are counters and flags that are cleared and not cleared when <ENRUN> bit is cleared to "0".
5-4	NR[1:0]	R/W	Noise filter 00:No filtering 01:Filters out pulses narrower than 31/fsys as noise (387.5ns@80MHz) 10:Filters out pulses narrower than 63/fsys as noise (787.5ns@80MHz) 11:Filters out pulses narrower than 127/fsys as noise (1587ns@80MHz) The digital noise filters remove pulses narrower than the width selected by <NR[1:0]>.
3	INTEN	R/W	Encoder interrupt enable 0:Disable 1:Enable <INTEN> enables or disables the ENC interrupt. Setting <INTEN> to "1" enables interrupt generation. Setting <INTEN> to "0" disables interrupt generation.
2-0	ENDEV[2:0]	R/W	Encoder pulse division factor 000:divided by 1 100:divided by 16 001:divided by 2 101:divided by 32 010:divided by 4 110:divided by 64 011:divided by 8 111:divided by 128 Sets encoder pulse division factor The frequency of the encoder pulse is divided by the factor specified by <ENDEV[2:0]>. The divided signal determines the interval of the event interrupt.

Note 1: In the encoder mode or timer mode, <P3EN> must be set to "0".
Note 2: If changing the mode, first read the flag to clear.

The operation mode has eight modes specified with<MODE[1:0]>, <P3EN> and <ZEN>.

The operation mode settings are as follows:

<MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Mode
00	0	0	A, B	Encoder mode
	1		A,B,Z	Encoder mode (use of Z)
01	0	0	U,V	Sensor mode (event count, 2-phase input)
		1	U,V,W	Sensor mode (event count, 3-phase input)
10	0	0	U,V	Sensor mode (timer count, 2-phase input)
		1	U,V,W	Sensor mode (timer count, 3-phase input)
11	0	0	-	Timer mode
	1		Z	Timer mode (use of Z)

The following is the status of <ENRUN> and corresponding signals.

Counter/flag	<ENRUN> = 0 (After reset)	<ENRUN> = 1 (Operating)	<ENRUN> = 0 (Stopping)	<ENRUN> = 0 Object flag/counter clear procedure
Encoder counter	0x000000	Count operation	Maintains a value when stopping	Software clear (<ENCLR> = 1 WR)
Noise filter counter	0y0000000	Count-up operation	Count-up operation (Always filtering)	Only reset
Encoder pulse division counter	0x00	Count-down operation	Stopped and cleared	Clear when <ENRUN> = "0"
Compare flag <CMP>	0	"1" is set when comparing Clear when read.	Cleared	Clear when <ENRUN> = "0"
Reverse error flag <REVERR>	0	"1" is set when error occurs. Clear when read.	Cleared	Clear when <ENRUN> = "0"
Z detection flag <ZDET>	0	"1" is set when Z is detected.	Cleared	Clear when <ENRUN> = "0"
Rotation direction bit <UD>	0	"0"/"1" is set depending on the direction	Cleared	Clear when <ENRUN> = "0"

14.4.3 ENxRELOAD(Encoder Counter Reload Register)

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

Bit	Bit Symbol	Type	Function
31-16	–	R	Read as "0".
15-0	RELOAD[15:0]	R/W	Sets the Encoder counter period (after multiplied by 4 or six) 0x0000 to 0xFFFF Z-phase is used : Sets the number of count pulses for one rotation Z-phase is not used : Sets the number of count pulses minus one for one rotation <RELOAD[15:0]> defines the encoder counter period multiplied by 4. If the encoder counter is configured as an up-counter, it increments up to the value programmed in <RELOAD[15:0]> and then wraps around to "0" on the next ENCLK. If the encoder counter is configured as a down-counter, it decrements to "0" and then is reloaded with the value of <RELOAD[15:0]> on the next ENCLK.

The RELOAD register is only used in Encoder mode.

14.4.4 ENxINT(Encoder Compare Register)

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

Bit	Bit Symbol	Type	Function		
31-24	–	R	Read as "0".		
23-0	INT[23:0]	R/W	Counter compare value setting		
			Encoder mode:	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = "1" is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = "1"is set, an interrupt request (INTENC0) occurs. However if <ZEN> = "1" is set, an interrupt request does not occur until <ZDET> = "1".	
			Sensor mode: (event count)	Interrupt condition of the encoder pulse position.	0x0000 to 0xFFFF
				While <CMPEN> = "1" is set, if an encoder counter value matches a value of <INT[15:0]>, <CMP> is set to "1". If <INTEN> = "1"is set, an interrupt request (INTENC0) occurs. This bit has no effect on a value of <ZEN>.	
			Sensor mode: (Timer count)	Interrupt condition of abnormal pulse detection time	0x000000 to 0xFFFFFF
				When <CMPEN> = "1" is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = "1" is set, an interrupt request (INTENC0) occurs. This bit has no effect on a value of <ZEN>.	
			Timer mode	Interrupt condition of timer compare	0x000000 to 0xFFFFFF
				When <CMPEN> = "1" is set, an internal counter value matches a value of <INT[23:0]>, abnormal pulse detection time error is determined and <CMP> is set to "1". If <INTEN> = "1" is set, an interrupt request (INTENC0) occurs. This bit has no effect on a value of <ZEN>.	

<INT[23:16]> is used only in Sensor mode (timer count) and Timer mode.

14.4.5 ENxCNT (Encoder Counter)

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

Bit	Bit Symbol	Type	Function		
31-24	–	R	Read as "0".		
23-0	CNT[23:0]	R/W	Encoder counter/capture value		
			Encoder mode:	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Encoder mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to the value of <RELOAD15:0>, it wraps around to "0" on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to "0", it is reloaded with the value of <RELOAD15:0> on the next ENCLK.	
			Sensor mode: (event count)	Counter value of encoder pulse	0x0000 to 0xFFFF
				The value of encoder count can be read. In Sensor Event Count mode, the encoder counter counts up or down on each encoder pulse (ENCLK). During CW rotation, encoder counter counts up; when it has reached to "0xFFFF", it wraps around to "0" on the next ENCLK. During CCW rotation, encoder counter counts down; when it has reached to "0", it wraps around to "0xFFFF" on the next ENCLK.	
			Sensor mode: (Timer count)	Pulse detection time or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read. In Sensor mode, the value of encoder counter can be read and captured by software on each encoder pulse (ENCLK) by writing "1" to <SFTCAP>. The captured value is cleared to "0" by system reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Sensor Timer Count mode, the encoder counter is configured as a free-running counter that counts up with fsys. The encoder counter is cleared to "0" when the encoder pulse (ENCLK) is detected. When it has reached to "0xFFFFFF", it wraps around to "0" automatically.	
			Timer mode	Capture value of internal counter or captured value by software	0x000000 to 0xFFFFFF
				The value of encoder counter can be read and captured by software by writing "1" to <SFTCAP>.When <ZEN> = "1", the value of the encoder counter is also captured into <CNT23:0> on the Z edge selected by <ZESEL>. The captured value is cleared to "0" by reset. It can also be cleared by clearing the counter by setting <ENCLR> to 1 and then setting <SFTCAP> to 1. In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached to "0xFFFFFF", it wraps around to "0" automatically.	

<CNT[23:16]> is used only in the sensor mode (Timer counting) or timer mode. In the encoder mode or sensor mode (event counting), always reads as "0".

14.5 Operational Description

14.5.1 Encoder mode

The high-speed position sensor determines the phase input from the AB encoder and the ABZ encoder.

- Event detection (rotation pulse) → interrupt generation
- Event count → match detection interrupt generation (measures the amount of transferring)
- Detects rotation direction
- Up/down-count (changeable in operation)
- Settable counter cycle

14.5.2 Sensor mode

The low-speed position sensor determines (zero-cross determination) the phase input from UV Hall sensor and UVW Hall sensor.

There are two kinds of sensor modes such as event count mode and timer count mode (counts with fsys).

14.5.2.1 Event Count Mode

- Event detection (rotation pulse) → interrupt generation
- Event count → match interrupt occurs (measuring the amount of transfer)
- Rotation direction detection

14.5.2.2 Timer count mode

- Event detection (rotation pulse) → interrupt generation
- Timer count
- Rotation direction detection
- Capture function → event capture (measures event intervals) → interrupt generation

  software capture
- Abnormal detection time error (timer compare) → match detection interrupt generation
- Reverse detection error → error flag caused by changing rotation direction

14.5.3 Timer mode

This mode can be used as a general-purpose 24-bit timer.

- 24-bit up counter
- Counter clear control (software clear, timer clear, external trigger and free-run count)
- Compare function → match detection interrupt generation
- Capture function → external trigger capture → interrupt generation

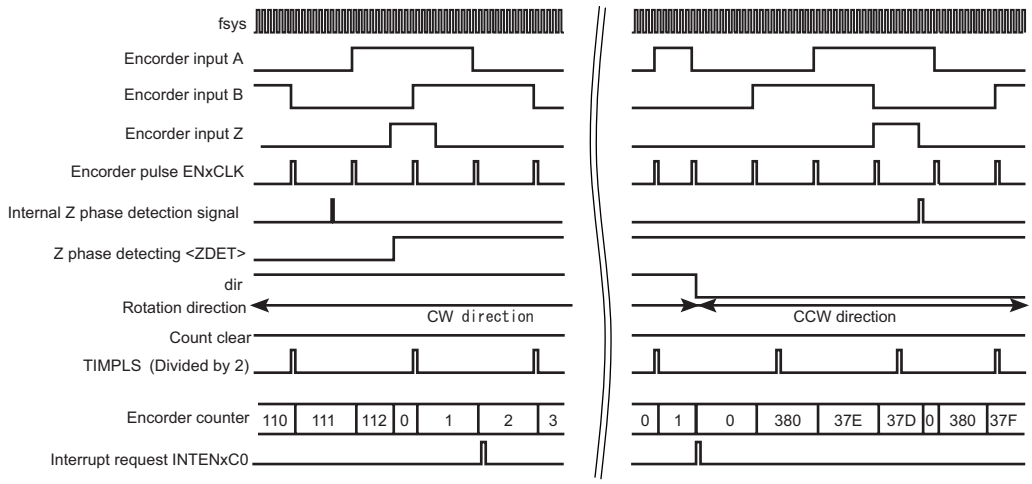
  software capture

14.6 Function

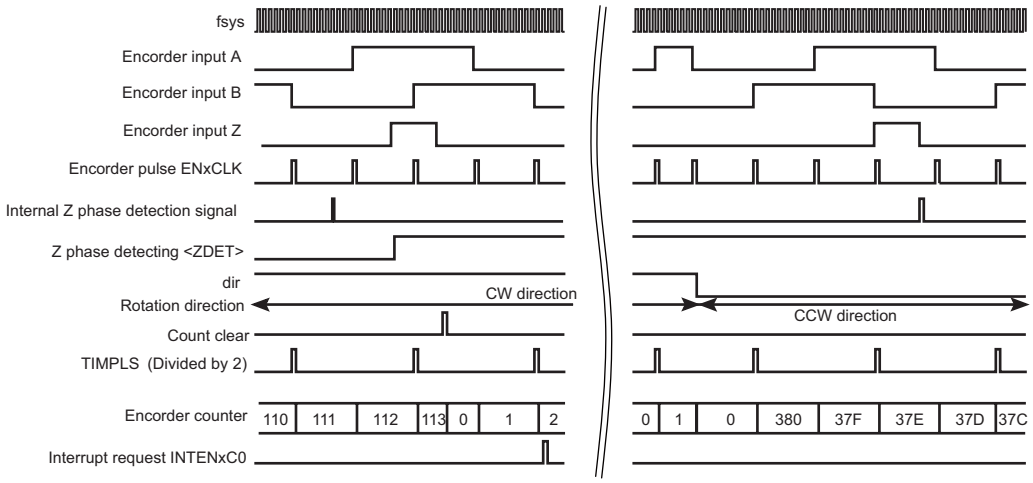
14.6.1 Mode operation outline

14.6.1.1 Encoder mode

1. If <ZEN> = 1 (<RELOAD> = 0x0380 , <EN0INT> = 0x0002)



2. If <ZEN> = 0 (<RELOAD> = 0x0380 , <EN0INT> = 0x0002)

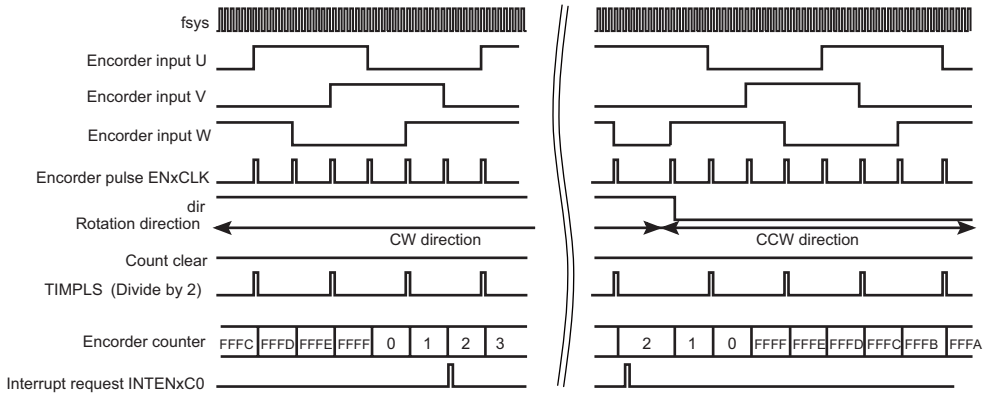


- The incremental encoder inputs of the MCU should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied by 4 clock derived from the decoded A and B quadrature signals.
- During CW rotation (i.e., A has the 90-degree phase lead to B), the encoder counter counts up; when it has reached to the value of <RELOAD>, it wraps around to "0" on the next ENCLK.
- During CCW rotation (i.e., A has the 90-degree phase lag to B), the encoder counter counts down; when it has reached to "0x0000", it is reloaded with the value of <RELOAD> on the next ENCLK.
- Additionally, when <ZEN> = "1", the encoder counter is cleared to "0" on the rising edge of Z during CW rotation and on the falling edge of Z during CCW rotation (at the internal Z_Detected timing). If the ENCLK edge matches Z edge, the encoder counter is cleared to "0" without incrementing or decrementing.

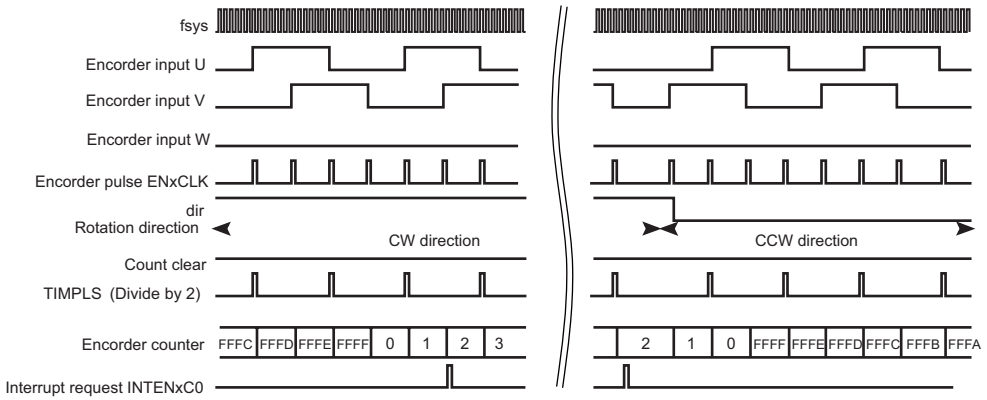
- When <ENCLR> is set to 1, the encoder counter is cleared to "0".
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of <EN0INT>. When <ZEN> = "1", however, an interrupt does not occur while <ZDET> = "0".
- When <ZDET> and <UD> are set to "0", <ENRUN> is cleared to "0".

14.6.1.2 Sensor mode (event count)

1. If <P3EN> = 1 (<EN0INT> = 0x0002)



2. If <P3EN> = 0 (<EN0INT> = 0x0002)

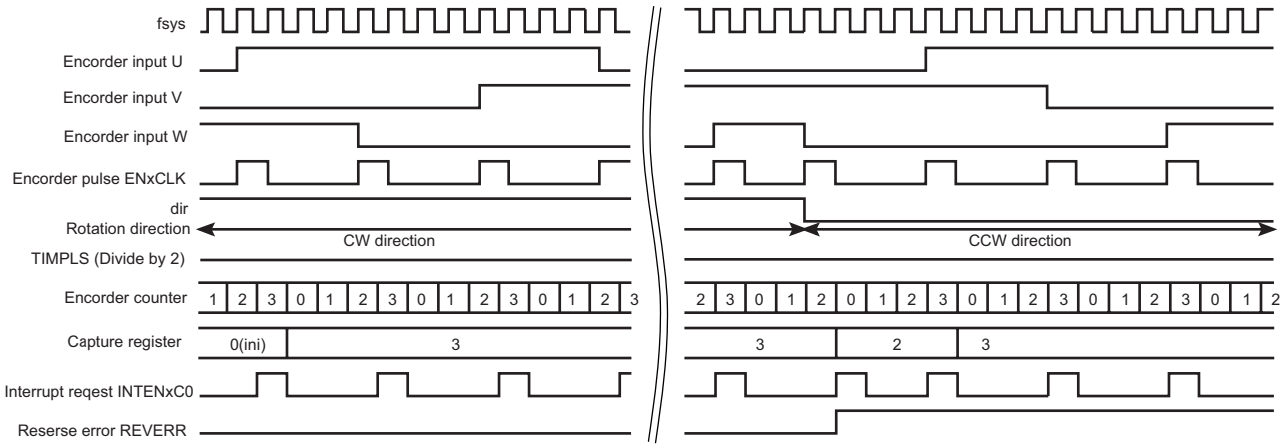


- The Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter counts the pulses of ENCLK, which is either multiplied by 4 clock (when <P3EN> = "0") derived from the decoded U and V signals or multiplied by 6 clock (when <P3EN> = "1") derived from the decoded U, V and W signals.
- During CW rotation (i.e., U channel has the 90-degree phase lead to V channel; V channel has the 90-degree phase lead to W channel), the encoder counter counts up; when it has reached to "0xFFFF", it wraps around to "0" on the next ENCLK.
- During CCW rotation (i.e., U channel has the 90-degree phase lag to V channel; V channel has the 90-degree phase lag to W), the encoder counter counts down; when it has reached to "0x0000", it wraps around to "0xFFFF" on the next ENCLK.
- When <ENCLR> is set to 1, the internal counter is cleared to "0".
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.

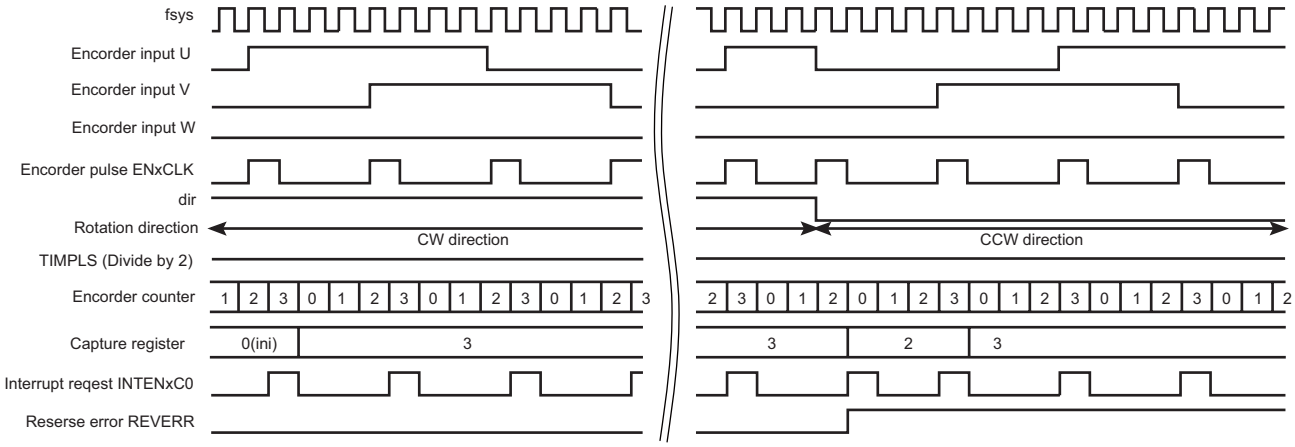
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the internal counter has reached to the value of <EN0INT>.
- When <UD> and <ENRUN> are set to "0", <UD> is cleared to "0".

14.6.1.3 Sensor mode (Timer count)

1. If <P3EN> = 1 (<EN0INT> = 0x0002)



2. If <P3EN> = 0 (<EN0INT> = 0x0002)

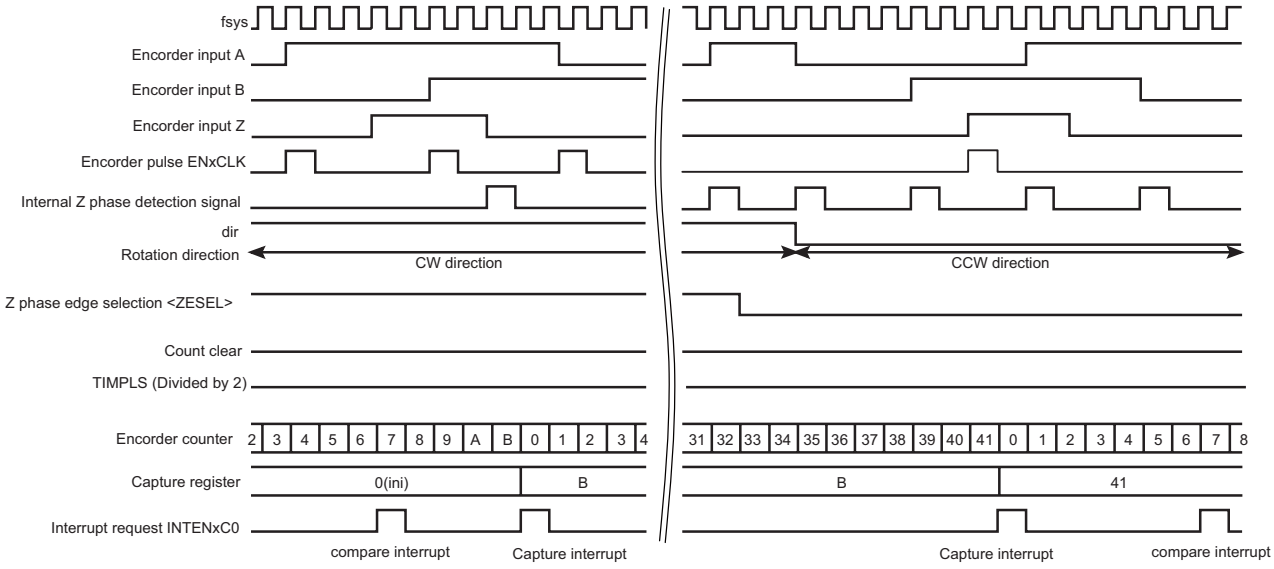


- In Sensor Timer Count mode, the Hall sensor inputs of the MCU should be connected to the U, V and W channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied by 4 clock (when <P3EN> = "0") derived from the decoded U and V signals or multiplied by 6 clock (when <P3EN> = "1") derived from the decoded U, V and W signals.
- The encoder counter always counts up; it is cleared to "0" on ENCLK. When the encoder counter has reached to "0xFFFFFF", it wraps around to "0".
- When <ENCLR> is set to 1, the encoder counter is cleared to "0".
- ENCLK captures the value of the encoder counter into the 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.

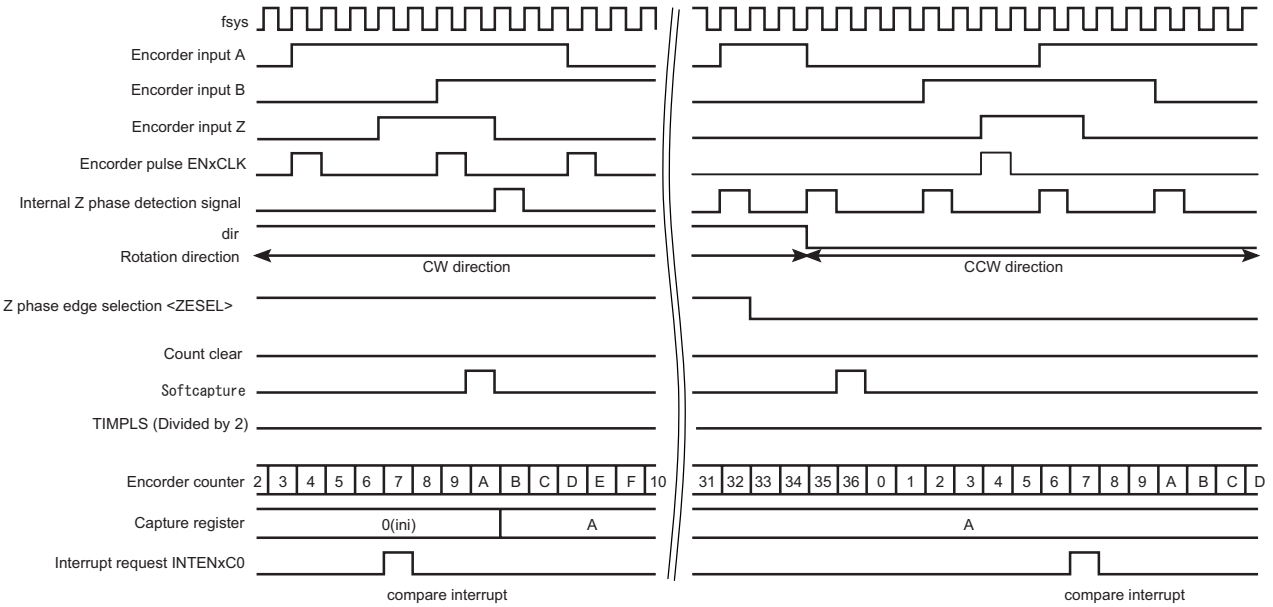
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of <EN0INT>.
- When <ENRUN> is set to "0", <UD> is cleared to "0".
- <REVERR> is set to 1 when the rotation direction has changed. This bit is cleared to "0" on a read.
- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

14.6.1.4 Timer mode

1. If <ZEN> = 1 (<EN0INT> = 0x0006)



2. If <ZEN> = 0 (<EN0INT> = 0x0006)

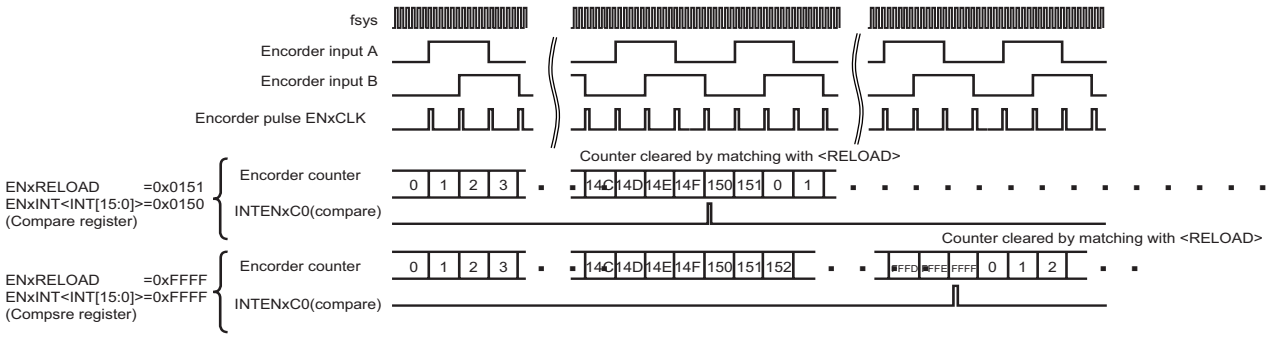


- When <ZEN> = "1", the Z input pin is used as an external trigger. When <ZEN> = "0", no external input is used to trigger the timer.

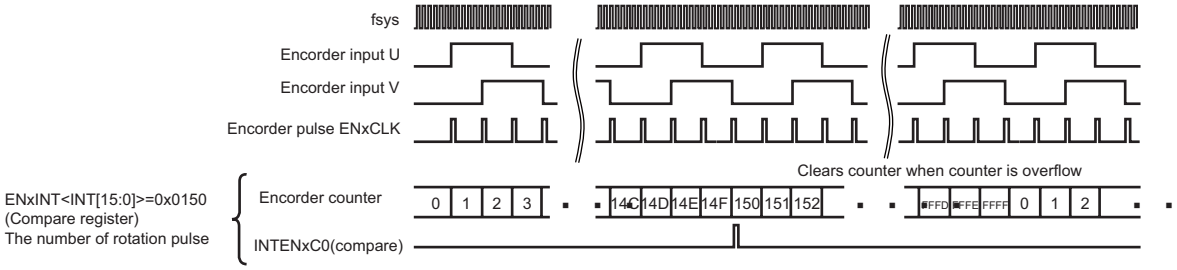
- The encoder counter always counts up. If <ZEN> = "1", the counter is cleared to "0" on the rising edge of Z when <ZESEL> is set to "0" and a falling edge when <ZESEL> is set to "1". When the encoder counter has reached to "0xFFFFFF", it wraps around to "0".
- When <ENCLR> is set to 1, the encoder counter is cleared to "0".
- Z-Detected causes the value of the encoder counter to be captured into the ENCNT register. The captured counter value can be read out of ENCNT.
- Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.
- <UD> is set to 1 during CW rotation and cleared to "0" during CCW rotation.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached to the value of <ENINT>.
- When <ENRUN> is set to "0", <UD> is cleared to "0".
- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

14.6.2 Counter and interrupt generate operation when <CMPEN> = 1

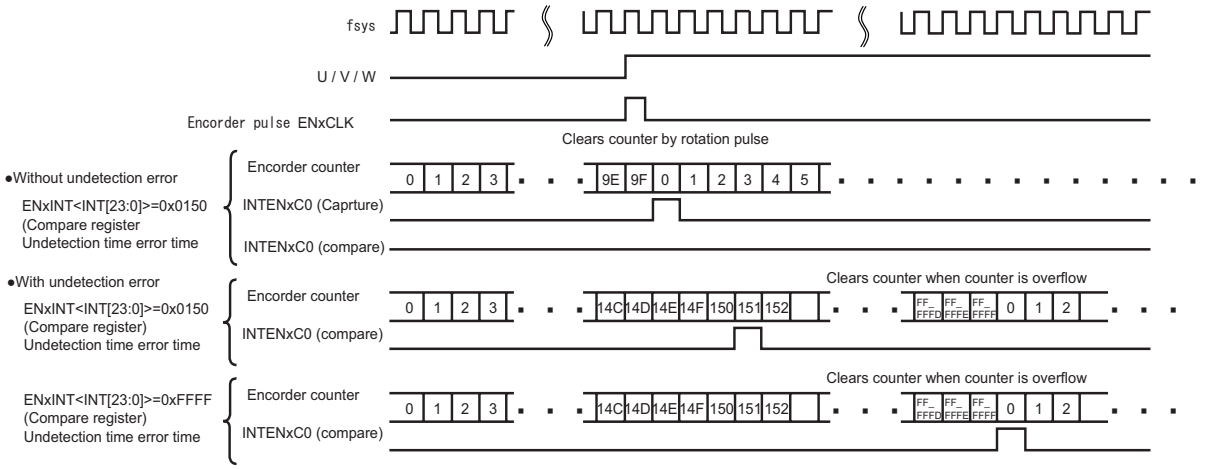
14.6.2.1 Encoder mode



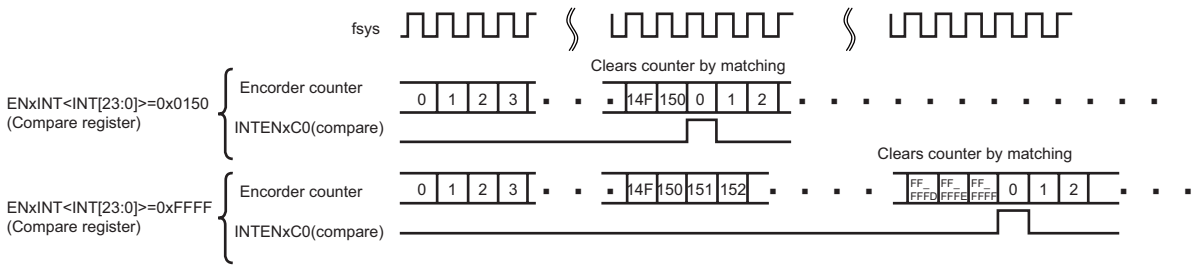
14.6.2.2 Sensor mode (event count)



14.6.2.3 Sensor mode (Timer count)



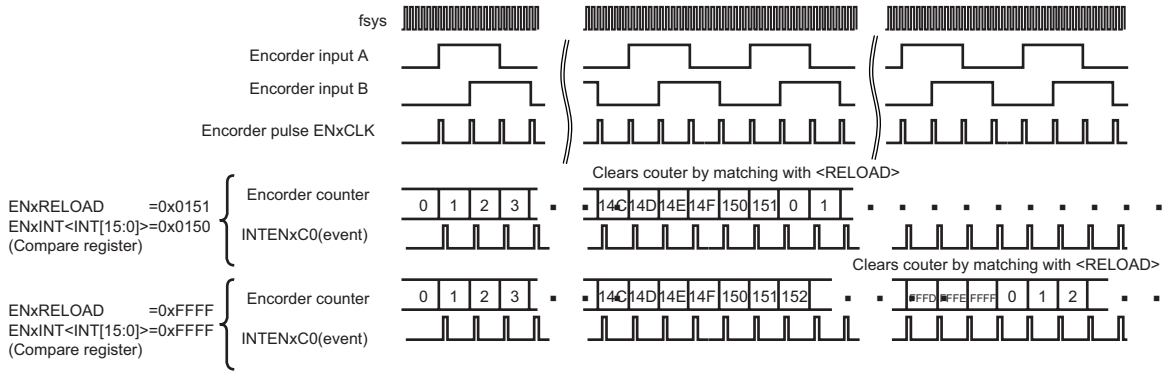
14.6.2.4 Timer mode



14.6.3 Counter and interrupt generate operation when <CMPEN> = 0

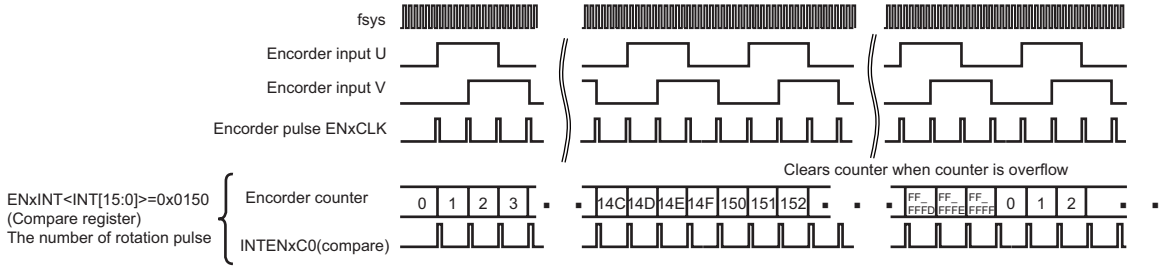
14.6.3.1 Encoder mode

<ENDEV>="000"

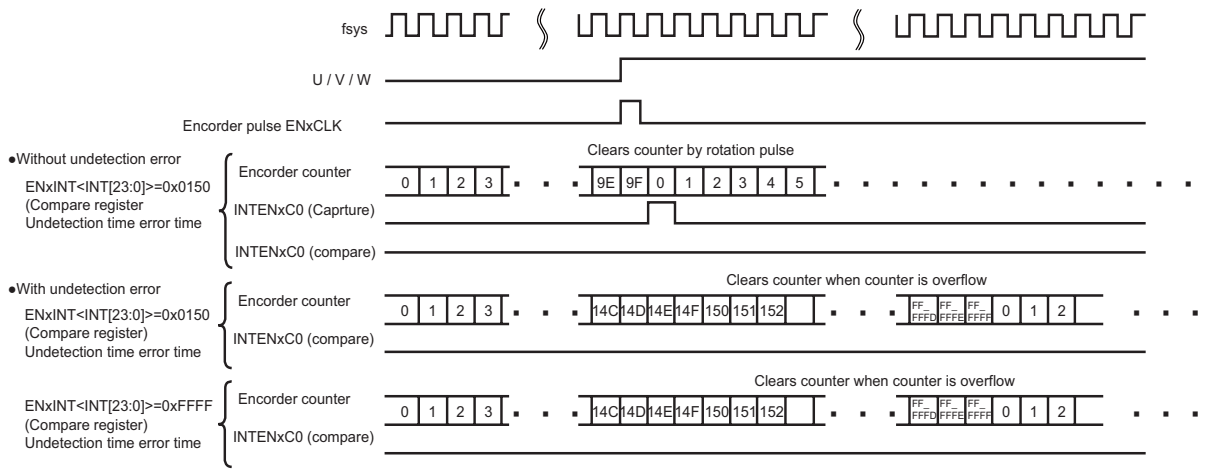


14.6.3.2 Sensor mode (event count)

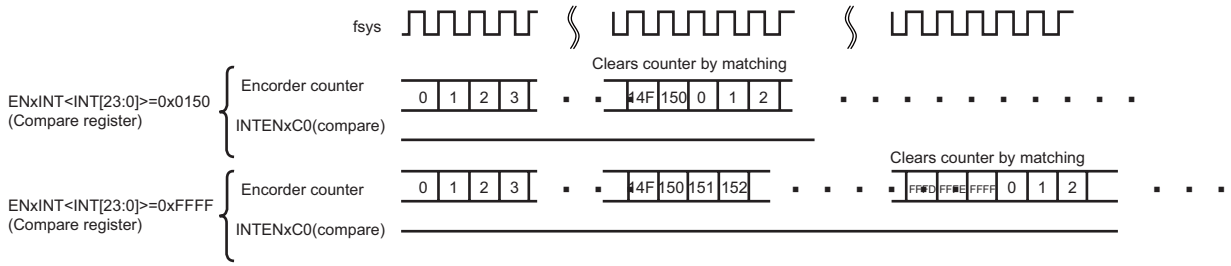
<ENDEV>="000"



14.6.3.3 Sensor mode (Timer count)



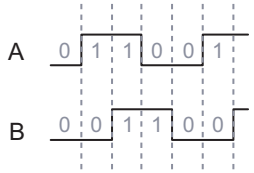
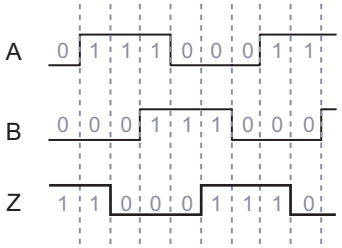
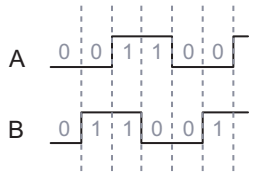
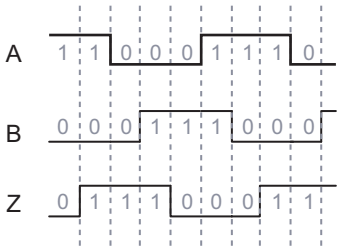
14.6.3.4 Timer mode



14.6.4 Encoder rotation direction

This circuit determines a phase either A-, B- or Z-phase.

It is used as 2-phase input (A,B) and 3-phase input (A,B,Z) in common. When 3-phase input is used, set <P3EN> = "1".

	2-phase input	3-phase input
CW direction		
CCW direction		

14.6.5 Counter Circuit

The counter circuit has a 24-bit up/down counter.

14.6.5.1 Operation Description

Depending on the operation modes, counting, clearing and reloading operation are controlled as described in Table 14-2.

Table 14-2 Counter control

Mode <MODE[1:0]>	<ZEN>	<P3EN>	Input pin	Count	Opera-tion	Counter clear condi-tion	Counter reload condition	Operational range of counter (Reload value)
Encoder mode 00	0	0	A,B	Encoder pulse (ENCLK)	UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD>	-	0x0000 to RELOAD>
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
	1		A,B,Z		UP	[1]<ENCLR> = 1 WR [2] Matches with <RELOAD> [3] Z-trigger	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (event count) 01	0	0	U,V		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	0x0000 to 0xFFFF
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
		1	U,V,W		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFF	-	
					DOWN	[1]<ENCLR> = 1 WR	[1] Matches with 0x0000	
Sensor mode (Timer count) 10	0	0	U,V	fsys	UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF	-	0x000000 to 0xFFFFFF
		1	U,V,W		UP	[3] Encoder pulse (ENCLK)	-	
Timer mode 11	0	×	-		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <EN0INT>	-	0x000000 to 0xFFFFFF
	1		Z		UP	[1]<ENCLR> = 1 WR [2] Matches with 0xFFFFFF [3] Matches with <EN0INT> [4] Z-trigger	-	

Note: The counter value is not cleared by writing "0" to <ENRUN>. If <ENRUN> = "1" is set again, the counter restarts from the counter value which has stopped. If clear the counter value, write "1" to <ENCLR> to execute software clear.

14.6.6 Interrupt

The interrupt consists of four interrupts including Event (divide pulse and capture), Abnormal detecting time, Timer compare and Capture interrupts.

14.6.6.1 Operational Description

When <INTEN> = "1" is set, interrupts occurs by counter value and encoder pulses.

Interrupt factor setting consists of six kinds setting with operation modes and the setting of <CMPEN> and <ZEN>.Table 14-3 shows interrupt factors.

Table 14-3 Interrupt factors

	Interrupt factor	Description	Mode	Interrupt output	Status flag
1	Event count interrupt	When <CMPEN> = 1, the encoder counter counts events (encoder pulses). When it has reached to the value programmed in <EN0INT>, an interrupt occurs.	Encoder mode and Sensor mode (event count)	<INTEN> = 1 and <CMPEN> = 1	<CMP>
2	Event interrupt (divide pulse)	An interrupt occurs on each divided clock pulse (1 to 128 divide), which is derived by dividing the encoder pulse by a factor programmed in <ENDEV>.		<INTEN> = 1	Not available
3	Event interrupt (capture interrupt)	An interrupt occurs to indicate that an event (encoder pulse) has occurred, causing the counter value to be captured on the rotation pulse timing.	Sensor mode (Timer count)	<INTEN> = 1	Not available
4	Abnormal detection time error interrupt	When <CMPEN> = 1, the ENC uses a counter that counts up with fsys and is cleared by an event (encoder pulse). If no event occurs for a period of time programmed in <EN0INT>, an interrupt occurs.		<INTEN> = 1 and <CMPEN> = 1	<CMP>
5	Timer compare interrupt	When <CMPEN> = 1, an interrupt occurs when the timer has reached to the value programmed in <EN0INT>.	Timer mode	<INTEN> = 1 and <CMPEN> = 1	<CMP>
6	Capture interrupt	An interrupt occurs when the counter value has been captured on an external trigger (Z input).		<INTEN> = 1	Not available

In Sensor Timer Count mode and Timer mode, the value of the encoder counter can be captured into the ENCNT register.

The captured counter value can be read out of the ENCNT register.

In Sensor Timer Count mode, the value of the encoder counter is captured into the ENCNT register upon occurrence of an event (encoder pulse). The counter value can also be captured by writing a 1 to <SFTCAP> by software.

In Timer mode, the counter value can be captured by writing a 1 to <SFTCAP> by software. If <ZEN> is set to 1, the counter value can also be captured by an edge of the Z signal input selected according to <ZESEL> by external trigger.

15. 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.

15.1 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.

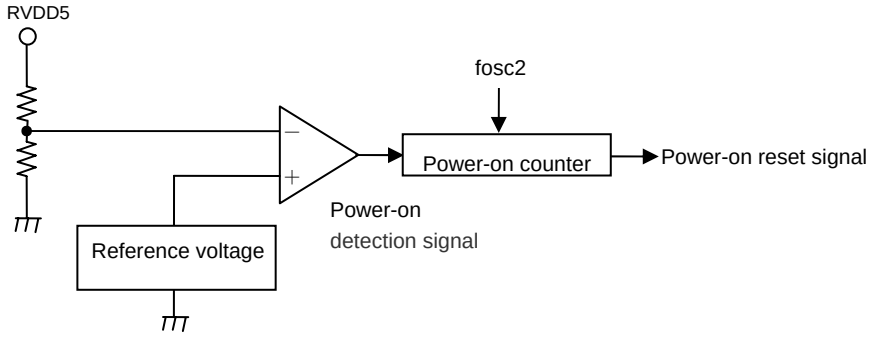


Figure 15-1 Power-on Reset Circuit

15.2 Function

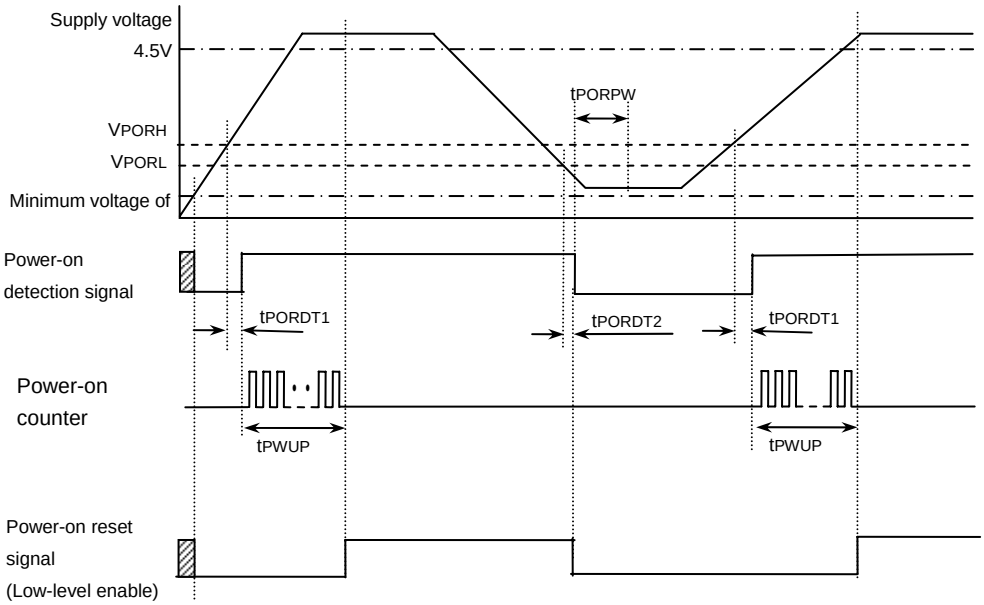
When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated. If the power supply voltage exceeds the releasing voltage of the power-on reset circuit, power-on counter is activated and $2^{15}/f_{osc2}$ (s) later, a power-on reset signal is released.

When power supply voltage goes down, if the supply voltage is equal to or lower than the detecting voltage of the power-on reset circuit, a power-on reset signal is generated.

During the generation of power-on reset, the power-on counter circuit, the CPU and peripheral circuits are reset.

When the power-on reset 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 TMPM376 cannot operate properly.

(Note) When the supply voltage rises, until the supply voltage (at RVDD5 pin) reach the recommended operating voltage range (4.5V through 5.5V) and 200μs 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 15-2 Operation Timing of Power-on Reset

Symbol	Parameter	Min	Typ.	Max	Unit
V_{PORH}	Power-on Reset releasing voltage	2.8	3	3.2	V
V_{PORL}	Power-on Reset detection voltage	2.6	2.8	3.0	V
t_{PORDT1}	Power-on Reset release response time		30		μs
t_{PORDT2}	Power-on Reset detection response time		30		μs
t_{PORPW}	Power-on Reset minimum pulse width	45			μs

Note 1 : Since the power-on reset releasing voltage and the power-on reset detection voltage relatively change, the detection voltage is never reversed.

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”.

16. Voltage Detection Circuit (VLTD)

The voltage detection circuit detects any decrease in the supply voltage and generates voltage detection reset signals

Note: The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (RVDD5). Refer to the electrical characteristics and take them into consideration when designing equipment.

16.1 Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (RVDD5) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDLVL), and the comparator compares it with the reference voltage.

When the supply voltage (RVDD5) becomes lower than the detection voltage (VDLVL), a voltage detection reset signal is generated.

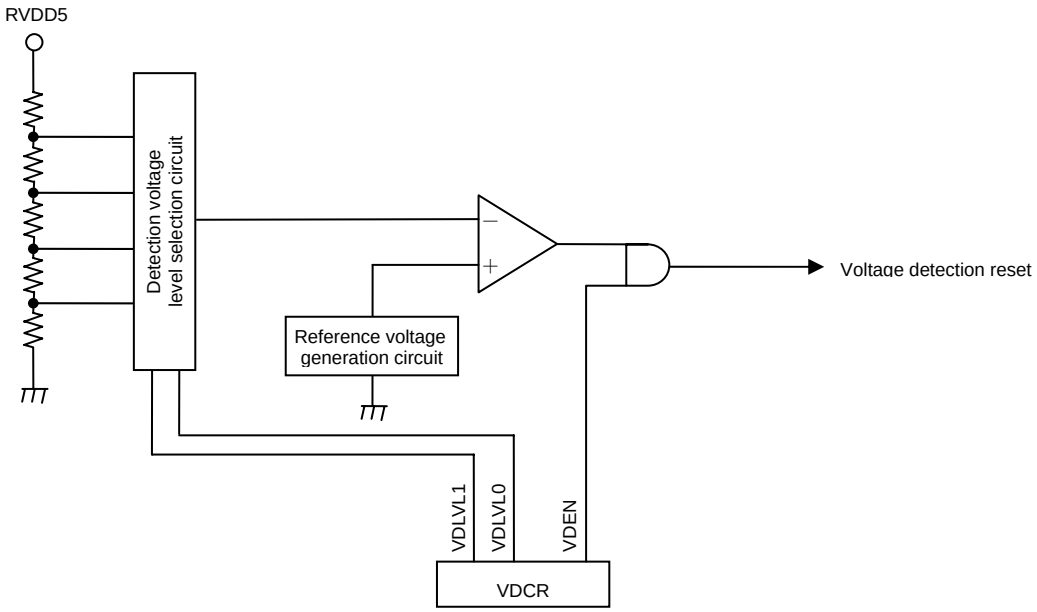


Figure 16-1 Voltage Detection Circuit

16.2 Control

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

Voltage detection control register

VDCR (0x4004_0900)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	VDLVL1	VDLVL0	VDEN
Read/Write	R	R	R	R	R	R/W		R/W
After reset	0	0	0	0	0	00		0

VDLVL[1:0]	Selection for detection voltage	00 : Reserved 01 : 4.1 ± 0.2 V 10 : 4.4 ± 0.2 V 11 : 4.6 ± 0.2 V
VDEN	Enables/disables the operation of voltage detection	0 : Disables the operation of voltage detection 1 : Enables the operation of voltage detection

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

16.3 Function

The detection voltage can be selected by VDCR<VDLVL[1:0]>. Enabling/disabling the voltage detection can be programmed by VDCR<VDEN>.

After the voltage detection operation is enabled, When the supply voltage (RVDD5) becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection reset signal is generated.

16.3.1 Enabling/disabling the voltage detection operation

Setting VDCR<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

VDCR<VDEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (RVDD5) is lower than the detection voltage (VDLVL), setting VDCR<VDEN> to "1" generates reset signal at the time.

16.3.2 Selecting the detection voltage level

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

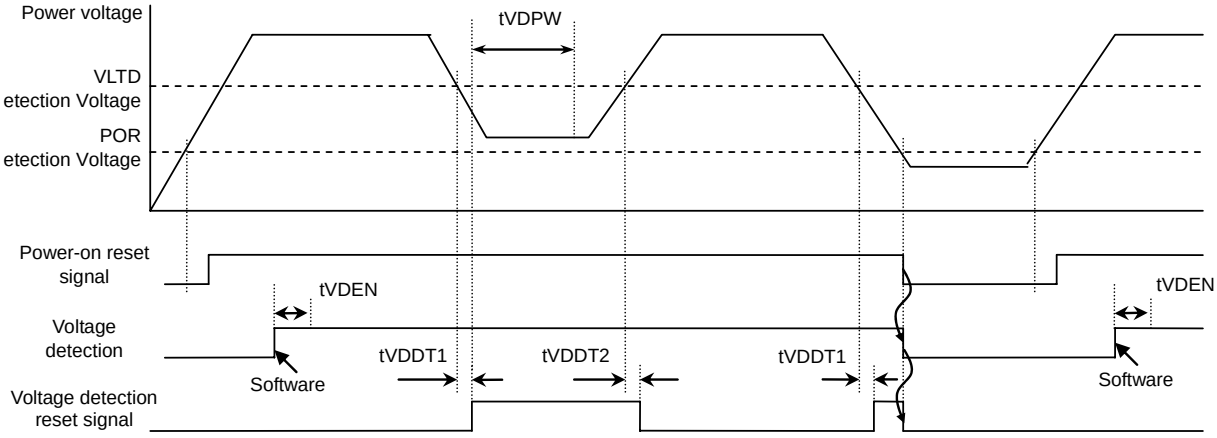


Figure 16-2 Voltage Detection Timing

Symbol	Parameter	Min	Typ.	Max	Unit
tVDEN	Setup time after enabling voltage detection		40		μs
tVDDT1	Voltage detection response time		40		μs
tVDDT2	Voltage detection releasing time		40		μs
tVDPW	Voltage detection minimum pulse width	45			μs

17. Oscillation Frequency Detector (OFD)

17.1 Configuration

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

The oscillation frequency detection is controlled by OFDCR1, OFDCR2 registers and the detection frequency range is specified by OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON which are the detection frequency setting registers. The lower detection frequency is specified by OFDMNPLLOFF/OFDMNPLLON registers and the higher detection frequency is specified by OFDMXPLLOFF/OFDMXPLLON registers.

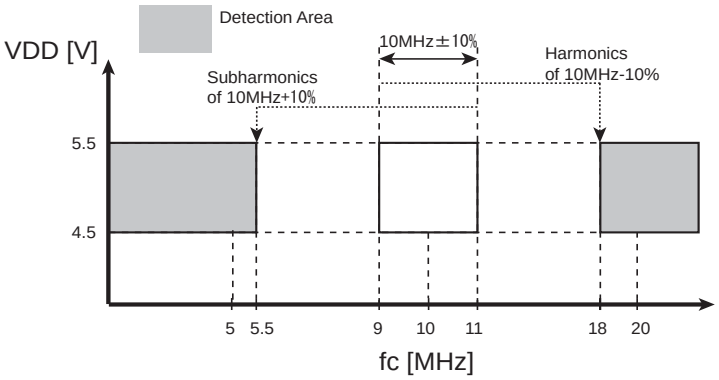
When the oscillation frequency detection is enabled, writing to OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON registers is disabled. Therefore, the setting the detection frequency to these registers should be done when the oscillation frequency detection is disabled. And writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON 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, power on reset or VLTD reset, write "0xF9" to OFDCR1 and write "0xE4" to OFDCR2 register to enable its function.

When the TMPM376FDDFG/FDFG detects the out of frequency by lower and higher detection frequency setting registers, all I/Os become high impedance by reset. In case of PLLOFF, OFDMNPLLOFF and OFDMXPLLOFF registers are valid for detection and the setting value of OFDMNPLLON/OFDMXPLLON registers are ignored. In case of PLLON, OFDMNPLLON and OFDMXPLLON registers are valid for detection and the setting value of OFDMNPLLOFF/OFDMXPLLOFF registers are ignored. By the oscillation frequency detection reset, all I/Os except power supply pins, $\overline{\text{RESET}}$, X1 and X2 become high impedance. If oscillation frequency detection reset is generated by detecting the stopping of high frequency, the internal circuities such as registers hold the condition at the timing of oscillation stop. To initialize these internal circuitries, an external re-starting of oscillation is needed.

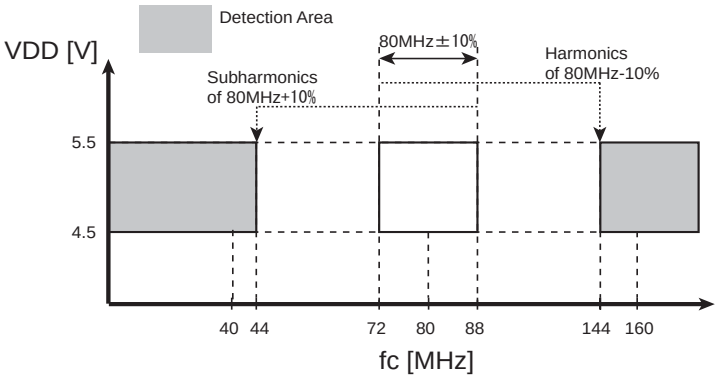
Since all registers for oscillation frequency detector (OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON) are not initialized by the reset generated from oscillation frequency detector, the detection of oscillation is keeps its function during the reset period of oscillation frequency detection. Therefore, if the oscillation frequency detection reset occurs, the reset is not released unless the CPU clock resumes its normal frequency.

Note 1: The oscillation frequency detection reset is available only in NORMAL and IDLE modes. In STOP mode, the oscillation frequency detection reset is disabled automatically.

Note 2: When the PLL is controlled (enabled or disabled) by the CGPLLSEL register, the OFD must be disabled beforehand. If OFD reset is generated with PLL-ON, the detection frequency setting registers (OFDMNPLLON/OFDMXPLLON) are automatically switched over to OFDMNPLLOFF/OFDMXPLLOFF.



(a) In case of PLLOFF



(b) In case of PLLON

Figure 17-1 Example of detection frequency range

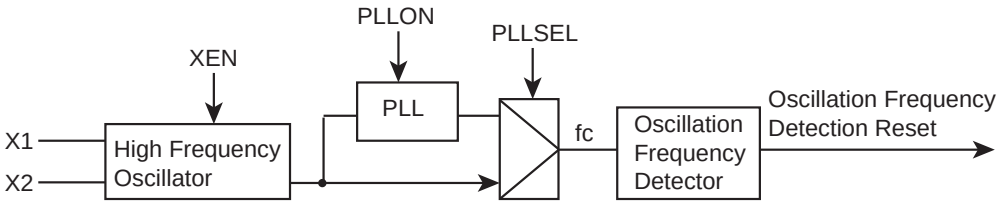


Figure 17-2 Oscillation Frequency Detector

17.2 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 (0x4004_0800)									
		31-8							
	Bit Symbol	-							
	Read/Write	R							
	After reset	0							
		7	6	5	4	3	2	1	0
	Bit Symbol	OFDWEN7	OFDWEN6	OFDWEN5	OFDWEN4	COFDWEN3	OFDWEN2	OFDWEN1	OFDWEN0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	1	1	0
	Function	0x06: Disabling of writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/ OFDMXPLLON (Write disable code) 0xF9: Enabling of writing to OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/ OFDMXPLLON (Write enable code) Others: Reserved (Note 1)							

Note 1: Only "0x06" and "0xF9" is valid to OFDCR1. If other value than "0x06" and "0xF9" is written to OFDCR1, "0x06" is written to OFDCR1 automatically.

Note 2: OFDCR1 is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

Oscillation frequency detection control register 2

OFDCR2 (0x4004_0804)									
		31-8							
	Bit Symbol	-							
	Read/Write	R							
	After reset	0							
		7	6	5	4	3	2	1	0
	Bit Symbol	OFDSEN7	OFDSEN6	OFDSEN5	OFDSEN4	OFDSEN3	OFDSEN2	OFDSEN1	OFDSEN0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	0x00: Disabling of oscillation frequency detection 0xE4: Enabling of oscillation frequency detection Others: Reserved (Note 1)							

Note 1: Only "0x00" and "0xE4" is valid to OFDCR2. Writing other value than "0x00" and "0xE4" to OFDCR2 is ignored.

Note 2: Writing to OFDCR2 is protected by setting "0x06" to OFDCR1 but reading from OFDCR2 is always enabled without setting of OFDCR1.

Note 3: OFDCR2 is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

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

OFDMNPLLOFF (0x4004_0808)									
		31-9						8	
	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	0	1	0

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

OFDMNPLLON (0x4004_080C)

	31-9						8	
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	0	1	0	0	0	1

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

OFDMXPLLOFF (0x4004_0810)

	31-9						8	
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	0	1	0	0	0

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

OFDMXPLLON (0x4004_0814)

	31-9						8	
Bit Symbol	-						OFDMXPLLON	
Read/Write	R						R/W	
After reset	0						1	
	7	6	5	4	3	2	1	0
Bit Symbol	OFDMXPLLON							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	1	1	1	0	0	1

Note 1: The after reset value is a tentative value.

Note 2: OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON can not be written when the oscillation frequency detection circuit is enabled (OFDCR2="0xE4") or writing is disabled with OFDCR1="0x06". An attempt to write OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON can not complete a write operation.

Note 3: Writing to OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON is protected by setting "0x06" to OFDCR1 but reading from OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON is always enabled without setting of OFDCR1.

Note 4: Specify an appropriate value to OFDMNPLLOFF and OFDMXPLLOFF depending on the clock frequency to be used under the condition of OFDMNPLLOFF<OFDMXPLLOFF. For how to calculate the value, refer to "17.3.2 Setting the Lower and Higher Frequency for Detection".

Note 5: Specify an appropriate value to OFDMNPLLON and OFDMXPLLON depending on the clock frequency to be used under the condition of OFDMNPLLON<OFDMXPLLON. For how to calculate the value, refer to "17.3.2 Setting the Lower and Higher Frequency for Detection".

Note 6: OFDMNPLLOFF, OFDMNPLLON, OFDMXPLLOFF and OFDMXPLLON are initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

Note 7: OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of PLLON.

17.3 Function

17.3.1 Enabling and Disabling the Oscillation Frequency Detection

Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection, and writing "0x00" to OFDCR2 with OFDCR1="0xF9" disables the oscillation frequency detection.

Registers of OFD are initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

Since OFDCR1 is initialized to "0x06" and OFDCR2 is initialized to "0x00" by resets shown above, oscillation frequency detection and writing to the registers are desabled. Reading from OFDCR2 is always enabled without setting of OFDCR1.

Note: After writing data to OFDCR2, set "0x06" to OFDCR1 to protect OFDCR2 register.

When STOP mode is executed with OFDCR2=0xE4, the oscillation frequency detection is automatically disabled. After releasing STOP and warming up period, the oscillation frequency detection is enabled. The oscillation frequency detection is available only in NORMAL and IDLE mode. Table 17-1 shows the availability of oscillation frequency detector.

Table 17-1 Availability of oscillation frequency detector

Operating Mode	Oscillation Frequency Detection (OFDCR2=0xE4)	All I/Os condition after Oscillation Frequency Detection RESET (Except power supply, $\overline{\text{RESET}}$, X1, X2 pins)
NORMAL	Available	High impedance
IDLE	Available	High impedance
STOP (Including warming up period)	Oscillation Frequency Detection is disabled automatically.	
Reset by oscillation frequency detection reset	Available	High impedance
Watchdog timer reset SYSRESETREQ reset	Available	High impedance
RESET by external reset power on reset VLTD reset	Disable	-

Figure 17-3 Availability of Oscillation Frequency Detection

17.3.2 Setting the Lower and Higher Frequency for Detection

The higher and lower limit of the detection frequency is calculated from the maximum error of the target clock and the reference. The reference clock frequency is 9.5 MHz and the error is ± 10%.

a)	target clock	Max.
b)		Min.
c)	reference clock	Max. (10.5MHz)
d)		Min. (8.5MHz)

How to calculate the setup value is shown below.

higher limit of the detection frequency	$= 1 \div \{ (d \div 2^7) \div (a \div 4) \}$	(truncate after the decimal places)
lower limit of the detection frequency	$= 1 \div \{ (c \div 2^7) \div (b \div 4) \}$	(round up after the decimal places)

17.3.3 Oscillation Frequency Detection Reset

If the TMPM376FDDFG/FDFG detects lower frequency specified by OFDMNPLLOFF/OFDMNPLLON or higher frequency specified by OFDMXPLLOFF/OFDMXPLLON, the oscillation frequency detector outputs a reset signal for all I/Os.

a. When the high frequency oscillation becomes abnormal

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

b. When the high frequency oscillation stops

When the high frequency oscillation stops for some period (T_{OFD}), the oscillation frequency detection reset is generated. By oscillation frequency detection reset initializes all I/Os except power supply pins, $\overline{\text{RESET}}$, 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.

18. Watchdog Timer(WDT)

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

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

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

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

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

18.1 Configuration

Figure 18-1shows the block diagram of the watchdog timer.

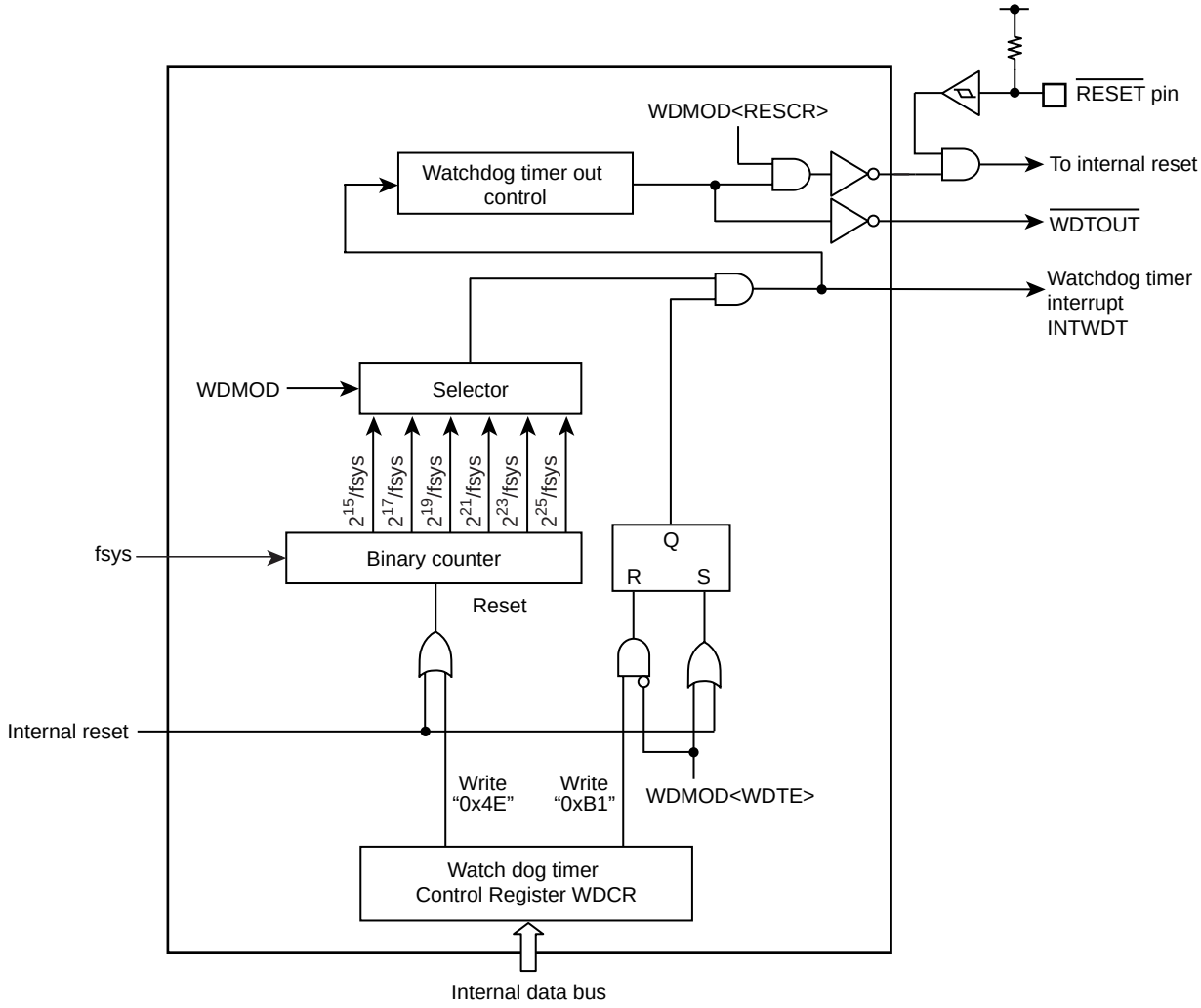


Figure 18-1 Block Diagram of the Watchdog Timer

18.2 Register

The followings are the watchdog timer control registers and addresses.

Base Address = 0x4004 _ 0000

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

18.2.1 WDMOD(Watchdog Timer Mode Register)

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

Bit	Bit Symbol	Type	Function
31-8	–	R	Read as 0.
7	WDTE	R/W	Enable/Disable control 0:Disable 1:Enable
6-4	WDTP[2:0]	R/W	Selects WDT detection time(Refer toTable 18-1) 000: 2^{15}/fsys 100: 2^{23}/fsys 001: 2^{17}/fsys 101: 2^{25}/fsys 010: 2^{19}/fsys 110:Setting prohibited. 011: 2^{21}/fsys 111:Setting prohibited.
3	–	R	Read as 0.
2	I2WDT	R/W	Operation when IDLE mode 0: Stop 1:In operation
1	RESCR	R/W	Operation after detecting malfunction 0: INTWDT interrupt request generates. (Note) 1: Reset
0	–	R/W	Write 0.

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

Table 18-1 Detection time of watchdog timer (fc = 80MHz)

Clock gear value CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.41 ms	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms
100 (fc/2)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
101 (fc/4)	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
110 (fc/8)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
111 (fc/16)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s

18.2.2 WDCR (Watchdog Timer Control Register)

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

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

18.3 Operations

18.3.1 Basic Operation

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

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

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

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

18.3.2 Operation Mode and Status

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

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

The watchdog timer cannot be used as the high-speed frequency clock is stopped. Before transition to below modes, the watchdog timer should be disabled.In IDLE mode, its operation depends on the WDMOD <I2WDT> setting.

- STOP mode

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

18.4 Operation when malfunction (runaway) is detected

18.4.1 INTWDT interrupt generation

In the Figure 18-2 shows the case that INTWDT interrupt generates (WDMOD<RESCR>="0").

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

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

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

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

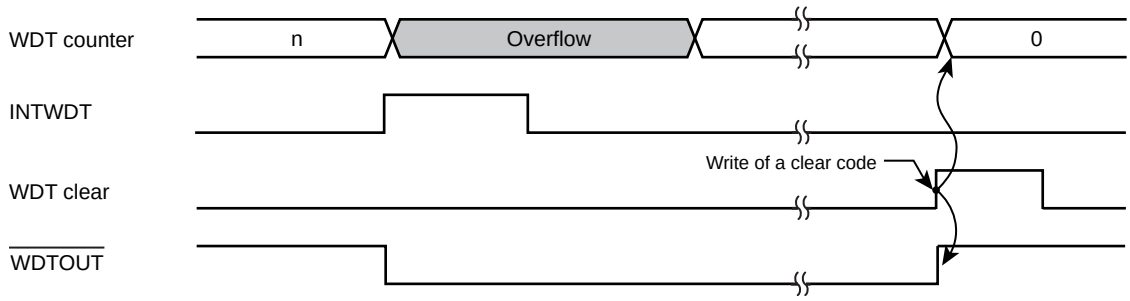


Figure 18-2 INTWDT interrupt generation

18.4.2 Internal reset generation

Figure 18-3 shows the internal reset generation (WDMOD<RESCR>="1").

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

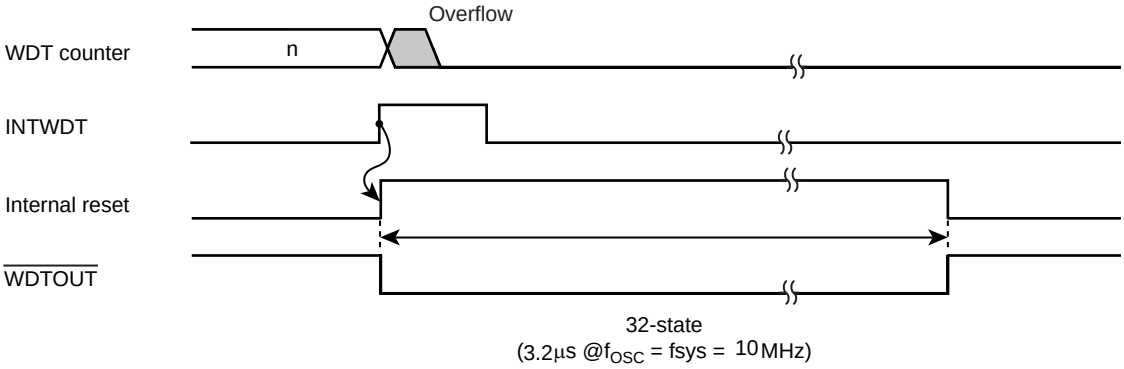


Figure 18-3 Internal reset generation

18.5 Control register

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

18.5.1 Watchdog Timer Mode Register (WDMOD)

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

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

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

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

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

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

3. Watchdog timer out reset connection <RESCR>

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

18.5.2 Watchdog Timer Control Register(WDCR)

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

18.5.3 Setting example

18.5.3.1 Disabling control

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

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

18.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

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

18.5.3.3 Watchdog timer clearing control

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

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

18.5.3.4 Detection time of watchdog timer

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

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

19. Flash

This section describes the hardware configuration and operation of the flash memory.

19.1 Flash Memory

19.1.1 Features

1. Memory capacity

TMPM376FDDFG/FDFG 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. TMPM376FDDFG/FDFG contains 128 words.

Page writing requires 1.25ms (typical) regardless of number of words.

A block erase requires 0.1 sec. (typical).

The following table shows write and erase time per chip.

Product Name	Memory size	Block Configuration				# of words	Write time	Erase time
		128 KB	64 KB	32 KB	16 KB			
TMPM376FDDFG / FDFG	512 KB	3	1	2	0	128	1.28 sec	0.4 sec

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 users.**

3. Programming method

There are two types of the onboard programming mode for users to program (rewrite) the device while it is mounted on the user's board:

a. User boot mode

The use’s original rewriting method can be supported.

b. Single boot mode

The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

4. Rewriting method

The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if a user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling / toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

5. Protect/ Security Function

This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See the chapter "ROM protection" for details of ROM protection and security function.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

19.1.2 Block Diagram of the Flash Memory Section

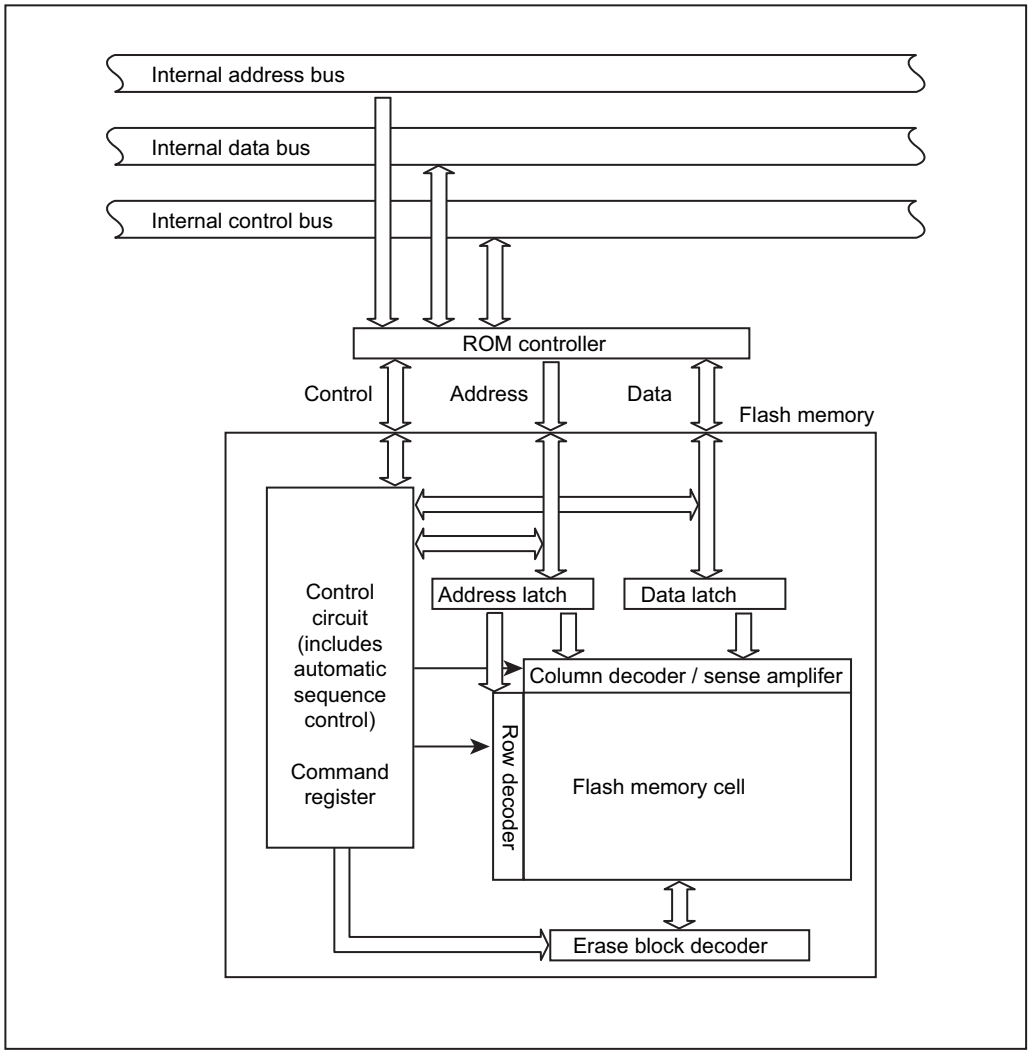


Figure 19-1 Block Diagram of the Flash Memory Section

19.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 19-1 Operation modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's set, are defined.The former is referred to as "normal mode" and the latter "user boot mode".
	User boot mode	A user can uniquely configure the system to switch between these two modes. For example, a user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0". A user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the 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 set.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ (PF0) pin while the device is in reset status.

Table 19-2 Operating Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PF0)
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

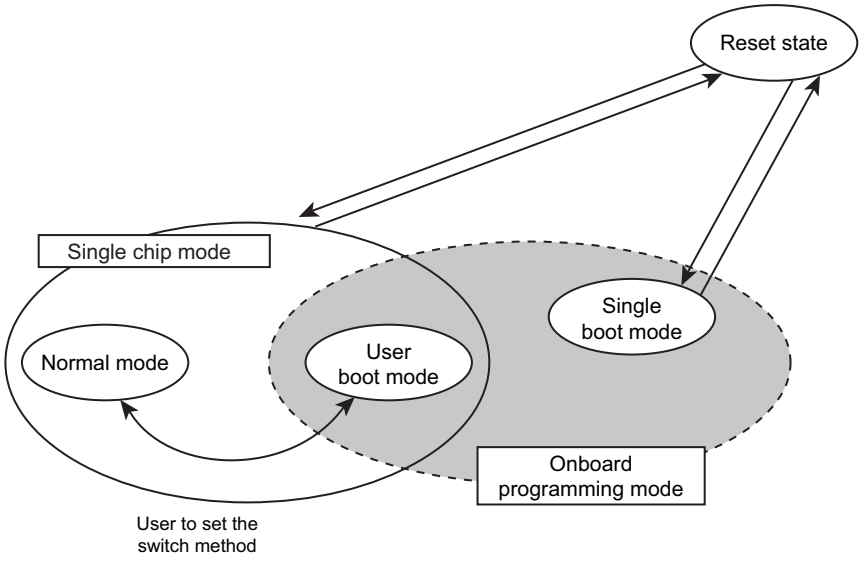


Figure 19-2 Mode Transition Diagram

19.2.1 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 (0.15μs with 80MHz operation; the "1/1" clock gear mode is applied after reset).

Note 1: **It is necessary to apply "0" to the RESET inputs upon power on for a minimum duration of 700 μs regardless of the operating frequency.**

Note 2: **While flash auto programming or erasing is in progress, at least 0.5 μs of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.**

19.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the user application.

The condition to switch the modes needs to be set by using the I/O of TMPM376FDDFG/FDFG in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete / writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. Be sure not to cause any exceptions including a non-maskable while User Boot Mode.

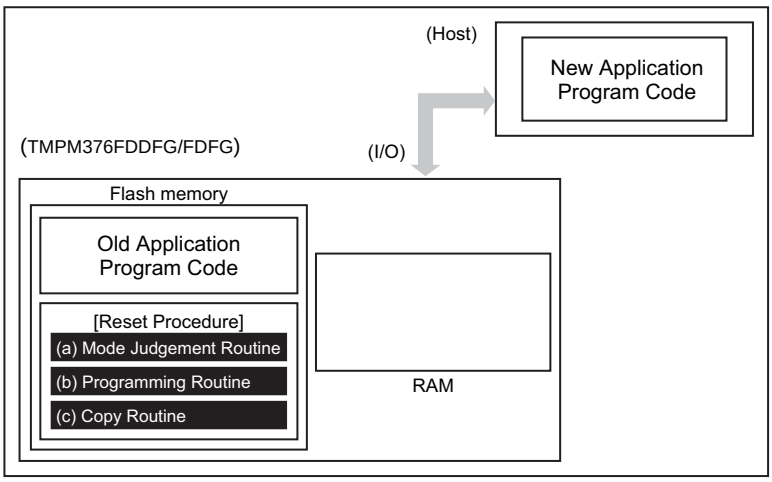
(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "19.3 On-board Programming of Flash Memory (Rewrite/Erase)".

19.2.2.1 (1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Step-1

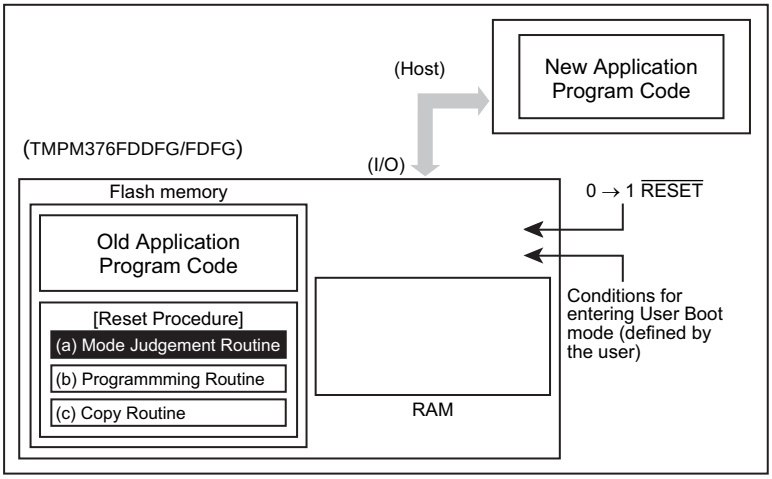
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM376FDDFG/FDFG on a printed circuit board, write the following program routines 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 TMPM376FDDFG/FDFG flash memory to either the TMPM376FDDFG/FDFG on-chip RAM or external memory device.



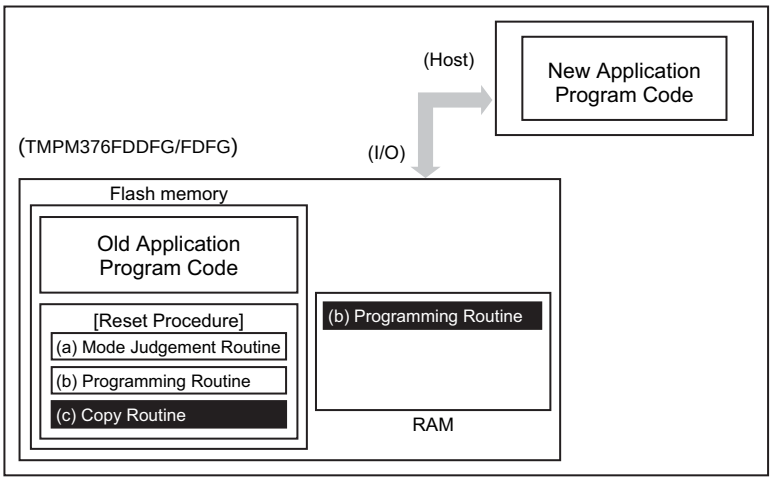
(2) Step-2

The following description is the case that programming routines are installed in the reset processing program. After $\overline{\text{RESET}}$ pin is released, the reset procedure determines whether to put the TMPM376FDDFG/FDFG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be not used while in User Boot mode.)



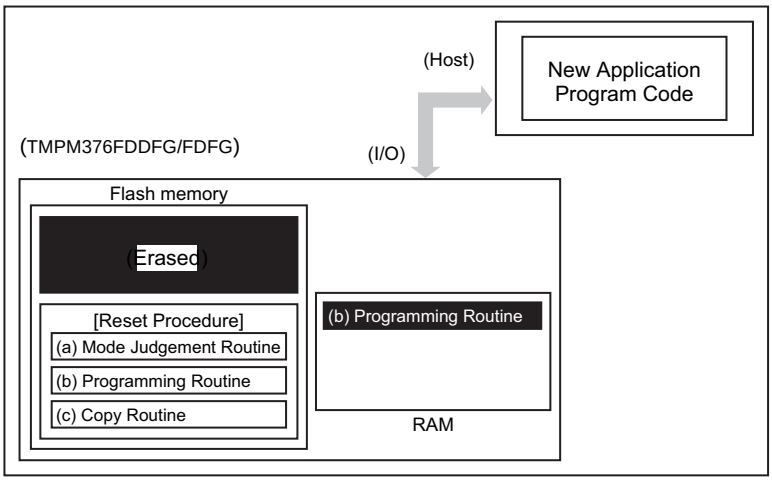
(3) Step-3

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM376FDDFG/FDFG on-chip RAM.



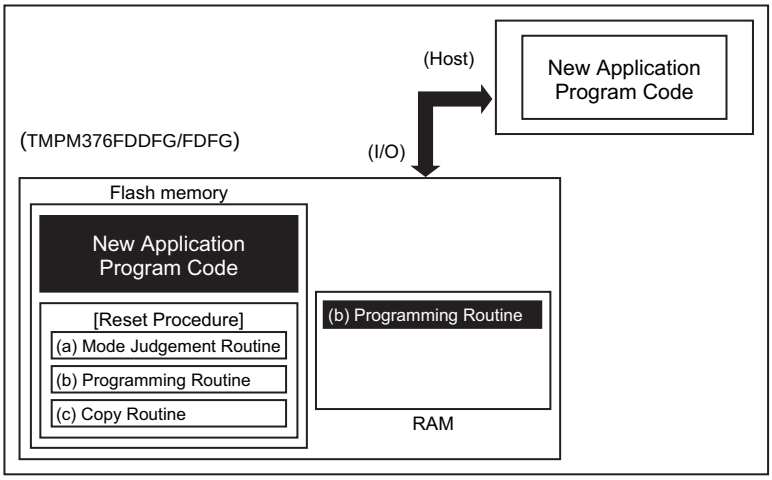
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to clear write or erase protection and erase a flash block containing the old application program code.



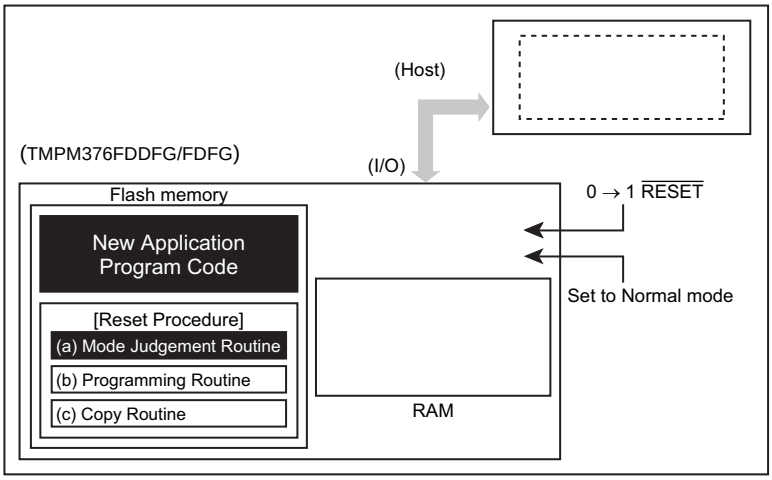
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM376FDDFG/FDFG. Upon reset, the on-chip flash memory is set to Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



19.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

(1) Step-1

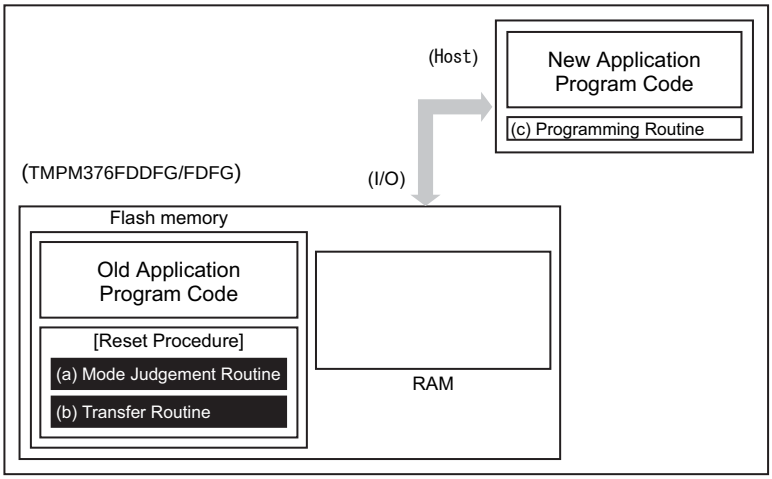
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM376FDDFG/FDFG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

(b) Transfer routine: Code to download new program code from a host controller

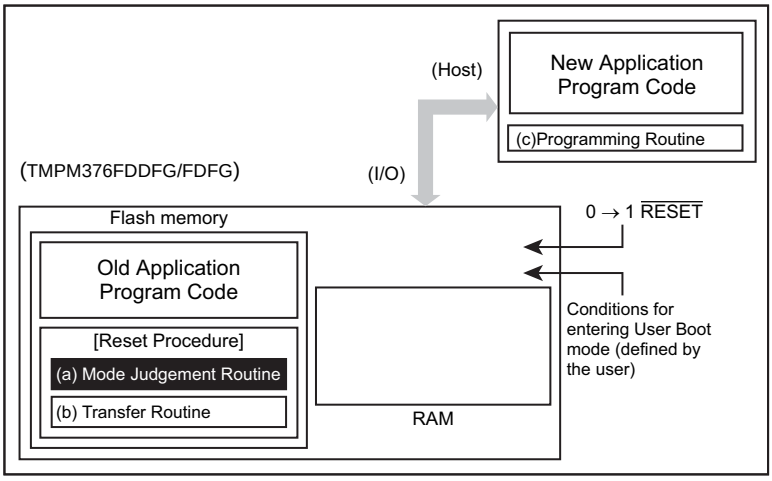
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory



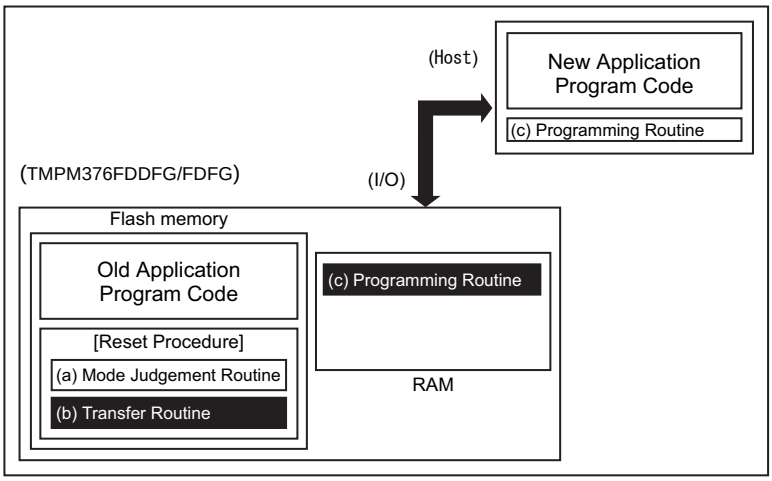
(2) Step-2

The following description is the case that programming routines are installed in the reset processing program. After RESET is released, the reset procedure determines whether to put the TMPM376FDDFG/FDFG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be not used while in User Boot mode).



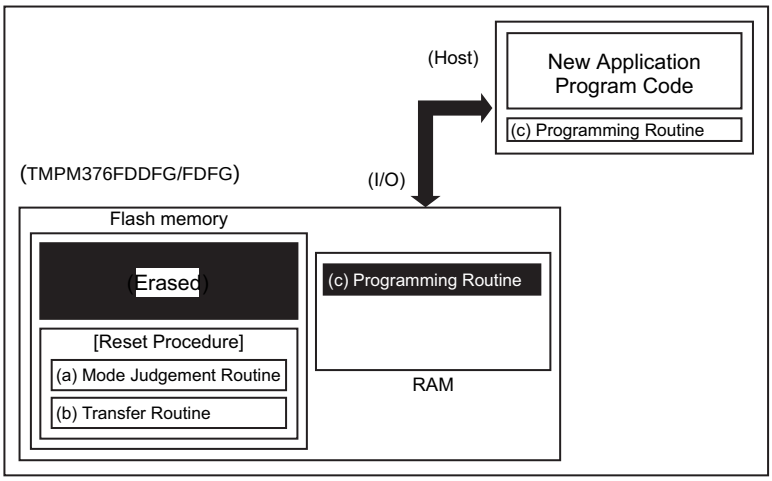
(3) Step-3

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM376FDDFG/FDFG on-chip RAM.



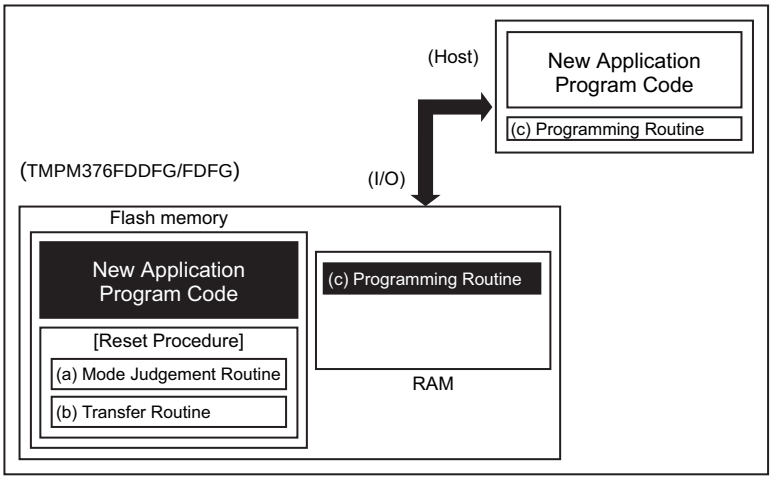
(4) Step-4

Jump program execution to the flash programming routine in the on-chip RAM to clear write or erase protection and erase a flash block containing the old application program code.



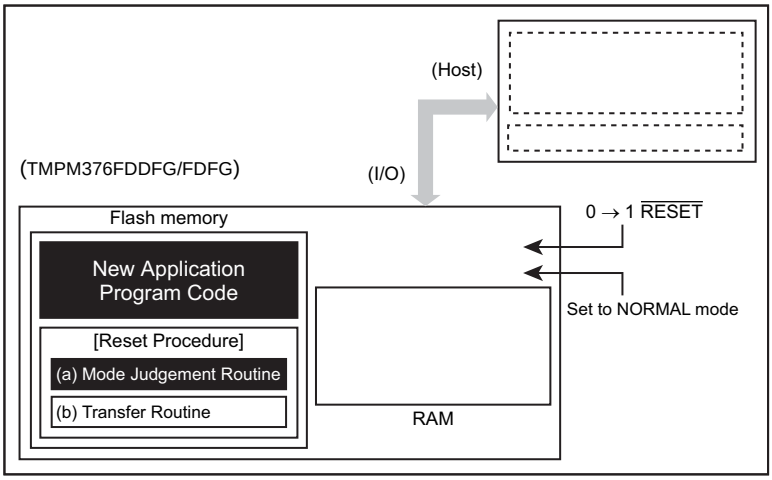
(5) Step-5

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user program area must be set.



(6) Step-6

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM376FDDFG/FDFG. Upon reset, the on-chip flash memory is set to Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



19.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM376FDDFG/FDFG 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.

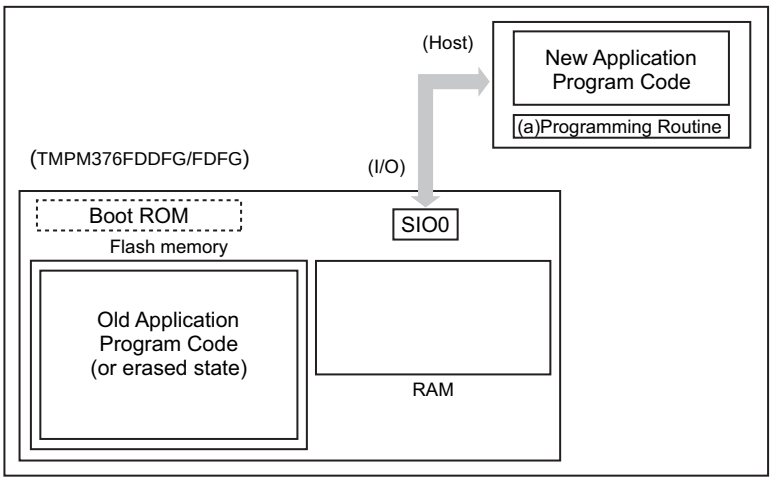
Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMPM376FDDFG/FDFG is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM376FDDFG/FDFG on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory. Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is verified before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted. As in the case of User Boot mode, all interrupts including the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is being erased or programmed. In Single Boot mode, the boot-ROM programs are executed in Normal mode.

Once re-programming is complete, it is recommended to set the write/erase protection to the relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

19.2.3.1 (2-A) Using the Program in the On-Chip Boot ROM

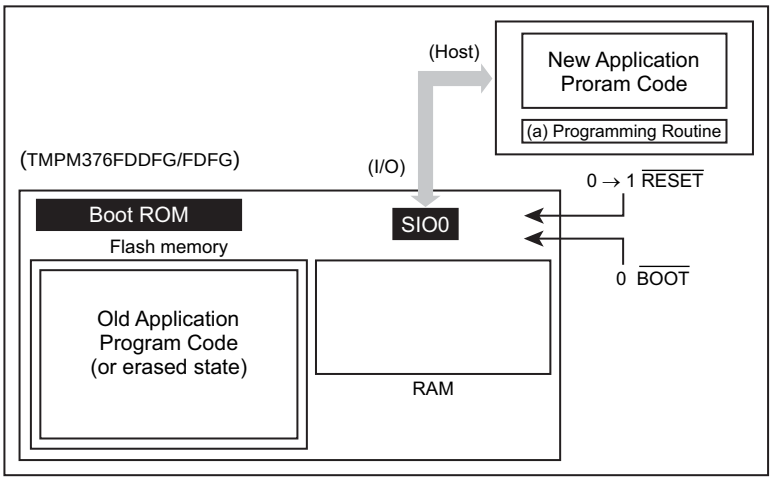
(1) Step-1

The flash block containing the old version of the program code does not need to be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.



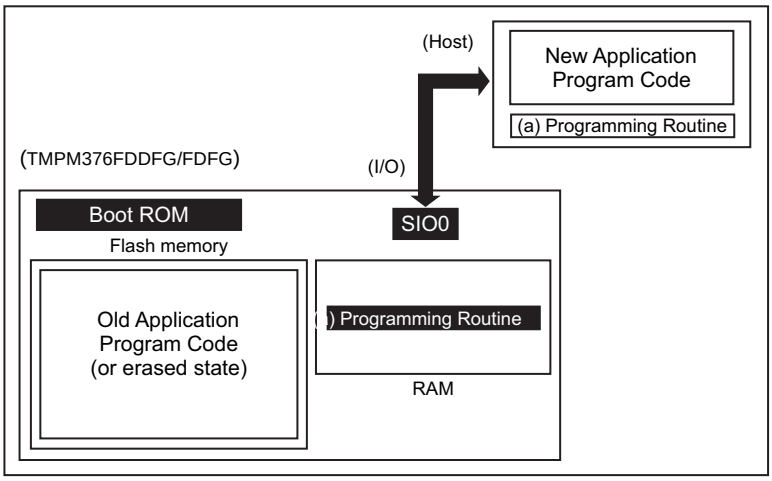
(2) Step-2

Set the $\overline{\text{RESET}}$ pin to "1" to cancel the reset of the TMPM376FDDFG/FDFG when the $\overline{\text{BOOT}}$ pin has already been set to "0". After reset, CPU reboots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is firstly compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFF).



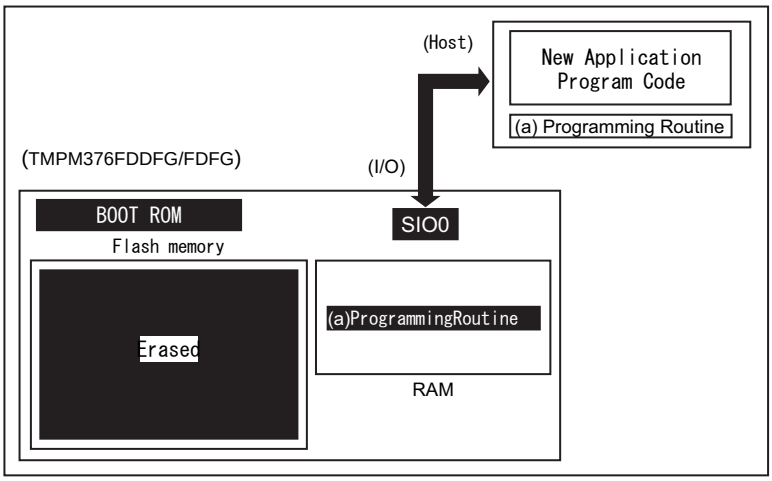
(3) Step-3

If the password is correct, the boot program downloads the programming routine (a) from the host controller into the on-chip RAM of the TMPM376FDDFG/FDFG. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



(4) Step-4

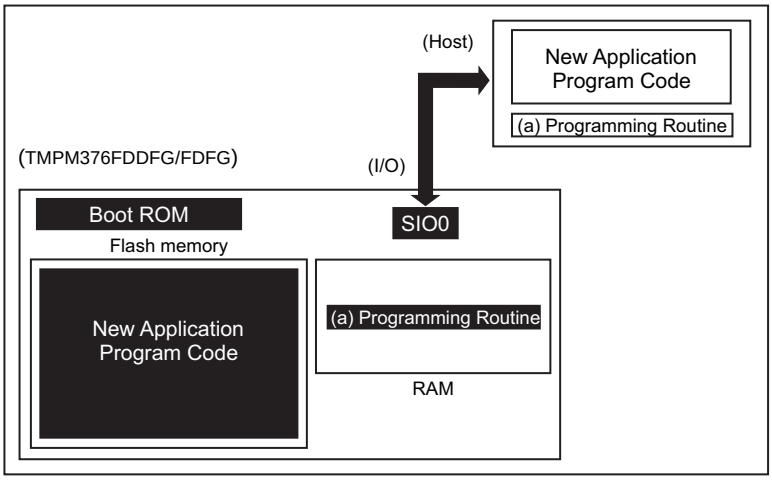
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



(5) Step-5

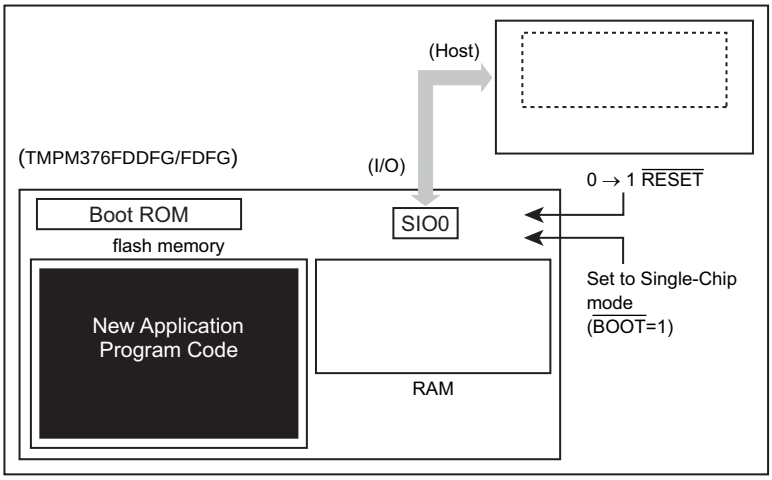
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. When the programming is completed, the writing or erase protection of that flash block in the user's program area must be set.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute in the on-chip RAM, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) Step-6

When programming of the flash memory is complete, power off the board and disconnect the cable between the host and the target board. Turn on the power again so that the TMPM376FDDFG/FDFG re-boots in Single-Chip (Normal) mode to execute the new program.



19.2.4 Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM376FDDFG/FDFG with Single Boot mode following the configuration shown below.

$\overline{\text{BOOT}}(\text{PF0}) = 0$
$\overline{\text{RESET}} = 0 \rightarrow 1$

Set the $\overline{\text{RESET}}$ input to "0", and set the each $\overline{\text{BOOT}}$ (PF0) pins to values shown above, and then release $\overline{\text{RESET}}$ pin (high).

19.2.5 Memory Map

Figure 19-3 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 and later addresses, and the Internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_0FFF.

The internal flash memory and RAM addresses of each device are shown below.

Product Name	Flash Size	RAM Size	Flash Address (Single Chip / Single Boot Mode)	RAM Address
TMPM376FDDFG/ FDFG	512 KB	32 KB	0x0000_0000 to 0x0007_FFFF 0x3F80_0000 to 0x3F87_FFFF	0x2000_0000 to 0x2000_7FFF

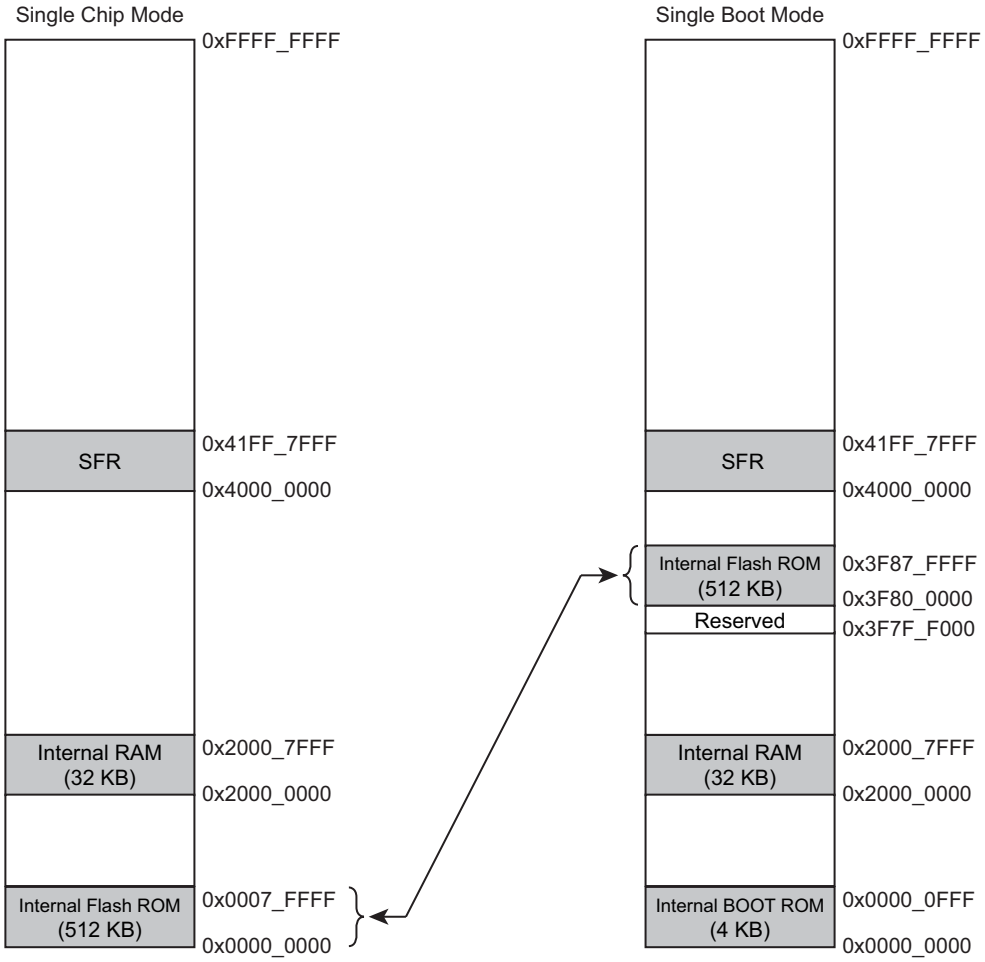


Figure 19-3 Memory Maps for TMPM376FDDFG/FDFG

19.2.6 Interface specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. The same configuration is applied to a communication format on a programming controller to execute the on-board programming. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below.

- UART communication

  Communication channel : SIO channel 0

  Serial transfer mode : UART (asynchronous) mode, half -duplex, LSB first

  Data length : 8 bits

  Parity bit : None

  STOP bit : 1 bit

  Baud rate : Arbitrary baud rate

- I/O Interface mode

  Communication channel : SIO channel 0

  Serial transfer mode : I/O interface mode, full -duplex, LSB first

  Synchronization clock (SCLK0) : Input mode

  Handshaking signal : PE4 configured as an output mode

  Baud rate : Arbitrary baud rate

Table 19-3 Required Pin Connections

Pin		Interface	
		UART	I/O Interface Mode
Power supply pins	DVDD5	o	o
	DVSS	o	o
	AVDD5A	o	o
	AVSSA	o	o
	AVDD5B	o	o
	AVSSB	o	o
	VOUT3	o	o
	VOUT15	o	o
	RVDD5	o	o
Mode-setting pin	$\overline{\text{BOOT}}$ (PF0)	o	o
Reset pin	$\overline{\text{RESET}}$	o	o
Communication pin	TXD0 (PE0)	o	o
	RXD0 (PE1)	o	o
	SCLK0 (PE2)	×	o (Input mode)
	PE4	×	o (Output mode)

19.2.7 Data Transfer Format

Table 19-4, Table 19-6 to Table 19-7 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to "19.2.10 Operation of Boot Program".

Table 19-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x40	Chip and protection bit erase

19.2.8 Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 19-5.

Table 19-5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	A program contained in the BOOT ROM uses the area, through 0x2000_0000 to 0x2000_03FF, as a work area. Store the RAM transfer program from 0x2000_0400 through the end address of RAM.
Internal ROM	The following addresses are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable. 0x3F87_FFF0 to 0x3F87_FFFF

19.2.9 Transfer Format for Boot Program

The following tables shows the transfer format for each Boot program command. Use this section in conjunction with Chapter "19.2.10 Operation of Boot Program".

19.2.9.1 RAM Transfer

Table 19-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM376FDDFG/FDFG	Baud rate	Data Transferred from the TMPM376FDDFG/ FDFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	–
	2 byte	–		ACK for the serial operation mode byte •For UART mode - Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) •For I/O Interface mode - Normal acknowledge :0x30
	3 byte	Command code (0x10)		–
	4 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	5 byte to 16 byte	Password sequence (12 bytes)) 0x3F87_FFF4 to 0x3F87_FFFF		–
	17 byte	Check SUM value for bytes 5 to 16		–
	18 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	19 byte	RAM storage start address 31 to 24		–
	20 byte	RAM storage start address 23 to 16		–
	21 byte	RAM storage start address 15 to 8		–
	22 byte	RAM storage start address 7 to 0		–
	23 byte	RAM storage start address 15 to 8		–
	24 byte	RAM storage start address 7 to 0		–
	25 byte	Check SUM value for bytes 19 to 24		–
	26 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	27 byte to mbyte	RAM storage data		–
	m+ 1 byte	Checksum value for bytes 27 to m		–
	m+ 2 byte	–		ACK for the checksum byte (Note 2) - Normal acknowledge : 0x10 - Negative acknowledge : 0xX1 - Communication error : 0xX8
RAM	m+ 3 byte	–		Jump to RAM storage start address

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

Note 3: **The 19th to 25th bytes must be within the RAM address range from 0x2000_0400 through the end address of RAM.**

19.2.9.2 Chip Erase and Protect Bit Erase

Table 19-7 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM376FDDFG/FDFG	Baud rate	Data Transferred from the TMPM376FDDFG/ FDFG to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode : 0x86 For I/O Interface mode : 0x30	Desired baud rate (Note 1)	–
	2 byte	–		ACK for the serial operation mode byte •For UART mode - Normal acknowledge : 0x86 (The boot program aborts if the baud rate can not be set correctly.) •For I/O Interface mode - Normal acknowledge :0x30
	3 byte	Command code (0x40)		–
	4 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x40 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	5 byte	Chip erase command code (0x54)		–
	6 byte	–		ACK for the command code byte (Note 2) - Normal acknowledge : 0x40 - Negative acknowledge : 0xX1 - Communication error : 0xX8
	7 byte	–		ACK for the chip erase command code byte - Normal acknowledge : 0x4F - Negative acknowledge : 0x4C
	8 byte	(Wait for the next command code.)		–

Note 1: **In I/O Interface mode, the baud rate for the transfers of the first and second byte must be 1/16 of the desired baud rate.**

Note 2: **In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.**

19.2.10Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these four commands, of which the details are provided on the following subsections.

1. RAM Transfer command

   The RAM Transfer command stores program code transferred from the host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the range from 0x2000_0400 to the end address of RAM, whereas the boot program area (0x2000_0000 to 0x2000_03FF) is unavailable. The user program starts at the assigned RAM address.

   The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 19.3. Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

Note: **If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

2. 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
TMPM376FDDFG/ FDFG	0x3F87_FFF0 to 0x3F87_FFF3

4. Flash Memory Chip Erase and Protection Bit Erase command

   This command erases the entire area of the flash memory automatically. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. 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.

19.2.10.1RAM Transfer Command

See Table 19-6 for the transfer format of this command.

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see "19.2.10.4 Determination of a Serial Operation Mode" described later. If the mode is determined as UART mode, the boot program checks if the baud rate setting can be performed. During the first-byte processing, receiving operation is prohibited. (SC0MOD0<RXE>=0)

   - To communicate in UART mode

     The 1st byte is set to "0x86" and is transmitted from the controller to the target board at the specified baud rate by setting UART. If the serial operation mode is determined as UART, then the boot program checks if the baud rate setting can be performed. If that baud rate cannot be set, the boot program aborts and any subsequent communications cannot be done. Please refer to "Baud rate setting" for the method of judging whether the setting of the baud rate is possible.

   - To communicate in I/O Interface mode

     The 1st byte is set to "0x30" and is transmitted from the controller to the target board at 1/16 of the desired baud rate by the synchronous setting. Same as the 1st byte, a 1/16 of the specified baud rate is used in the 2nd transmission. From the 3rd byte (operation command data), users can transmit data at specified baud rate.

     In I/O interface mode, CPU considers the reception terminal to be an input port and monitors the level of I/O port. If the baud rate is high or operation frequency is high, CPU may not distinguish the level of I/O port. To avoid this situation, the baud rate is set at the 1/16 of desired baud rate in the I/O interface. When the serial operation mode is determined as I/O Interface mode, SCLK Input mode is set. The controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no error acknowledge responce (bit 3, 0x08).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte where the serial operation mode is set. When 1st byte is determined as UART and can be set at the specified baud rate, data "0x86" is transmitted. When 1st byte is determined as I/O interface, data "0x30" is transmitted.

   • UART mode

   The 2nd byte is used for distinguishing whether the baud rate can be set. If the baud rate can be set, a value of SC0BRCR is renewed and data "0x86" is sent to the controller. If the baud rate cannot be set, transmit operation is stopped and no data is transmitted. After transmission of 1st byte completed, the controller allows for five seconds of time-out. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. Receiving operation is permitted by setting SC0MOD0<RXE>=1, before loading 0x86 to the SIO transmit buffer.

   • I/O Interface mode

   The boot program sets a value of the SC0MOD0 and SC0CR registers to configure the the I/O Interface mode and writes 0x30 to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. After the transmission of the 1st byte completed, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 of the desired baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller regards it as communication possible. From the 3rd byte, users can transmit data at specified baud rate. Receiving operation is permitted by setting SC0MOD0<RXE>=1, before loading 0x86 to the SIO.

3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there is a receive error, the boot program transmits 0xX8 (bit 3) and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 19-4, the boot program echoes it back to the controller. When the RAM Transfer command is received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in the later Section "Password". If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5th byte. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Area
TMPM376FDDFG/ FDFG	0x3F87_FFF4 to 0x3F87_FFFF

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, ignore the carries and caluculate the 8-bit two's complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in details in the later Section "Checksum Calculation".

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th byte. If there is a receive error, the boot program sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

   Next, the RAM Transfer routine performs the checksum operation to ensure 17th byte data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

   Finally, the password verification result is checked. If the following case is generated, the boot program transmits an acknowledge response (bit 0, 0x11) as a password error and waits for next operation command (3rd byte).

   - Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
   - Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

   When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31 to 24 of the address and the 22nd byte corresponds to bits 7 to 0 of the address.

   The start address of the stored RAM must be even address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15 to 8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7 to 0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, ignore the carries and caluculate the 8-bit two's complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in detail in the later Section "19.2.10.6 Checksum Calculation".

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there is a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 24th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

    - The 19th to 25th bytes data must be within the range of 0x2000_0400 to the end address of RAM.

    When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMPM376FDDFG/FDFG. Storage begins at the address specified by the 19th to 22nd bytes and continues for the number of bytes specified by the 23rd to 24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, ignore the carries and calculate the 8-bit two’s complement by using lower 8 bits then transmit this checksum value from the controller. The checksum calculation is described in detail in later Section "19.2.10.6 Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes. First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there is a receive error, the RAM Transfer routine sends back 0x18 (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., 1). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been completed successfully, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes.

19.2.10.2Chip and Protection Bit Erase Command

See Table 19-7 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.
2. From the Controller to the TMPM376FDDFG/FDFG

The 3rd byte, which the target board receives from the controller, is a command. The code for the Chip and protection bit erase command is 0x40.

3. From TMPM376FDDFG/FDFG to the Controller

   The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 3rd byte is equal to any of the command codes listed in Table 19-4, the boot program echoes it back to the controller. When the Chip and protection bit erase command was received, the boot program echoes back a value of 0x40. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. From the controller to the TMPM376FDDFG/FDFG

   The 5th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).

5. From TMPM376FDDFG/FDFG to the Controller

   The 6th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th byte.

   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 5th byte is equal to any of the command codes to enable erasing, the boot program echoes it back to the controller. When the Chip and Protection Erase command was received, the boot program echoes back a value of 0x54 and then branches to the Chip Erase routine. If the 5th byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

6. From TMPM376FDDFG/FDFG to the Controller

   The 7th byte indicates whether the Chip Erase command is normally completed or not.

   At normal completion, completion code (0x4F) is sent.

   When an error was detected, error code (0x4C) is sent.

7. The 9th byte is the next command code.

19.2.10.3Acknowledge Responses

The boot program represents processing states with specific codes. Table 19-8 to show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. The 3rd bit indicates a receive error. The 0th bit indicates an invalid command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not done in I/O Interface mode.

Table 19-8 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

Note: **In the UART mode, if the baud rate setting cannot be set, the communication is stopped without any response.**

Table 19-9 ACK Response to the Command Byte

Return Value	Meaning
0x?8 (See note)	A receive error occurred while receiving a command code.
0x?1 (See note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.
0x40	The Chip Erase command was received.

Note: **The upper four bits of the ACK response are the same as those of the previous command code.**

Table 19-10 ACK Response to the Checksum Byte

Return Value	Meaning
0xN8 (See note)	A receive error occurred.
0xN1 (See note)	A checksum or password error occurred.
0xN0 (See note)	The checksum was correct.

Note: **The upper four bits of the ACK response are the same as those of the operation command code. For example, it is 1 (N ; RAM transfer command data [7:4]) when password error occurs.**

Table 19-11 ACK Response to Chip and Protection Bit Erase Byte

Return Value	Meaning
0x54	The Chip Erase enabling command was received.
0x4F	The Chip Erase command was completed.
0x4C	The Chip Erase command was abnormally completed.

19.2.10.4Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must firstly send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 of the desired baud rate. Figure 19-4 shows the waveforms for the first byte in each mode.

Note: Between each point of A/B/C/D of Figure 19-4 is expressed as tAB, tAC, tAD, and tCD.

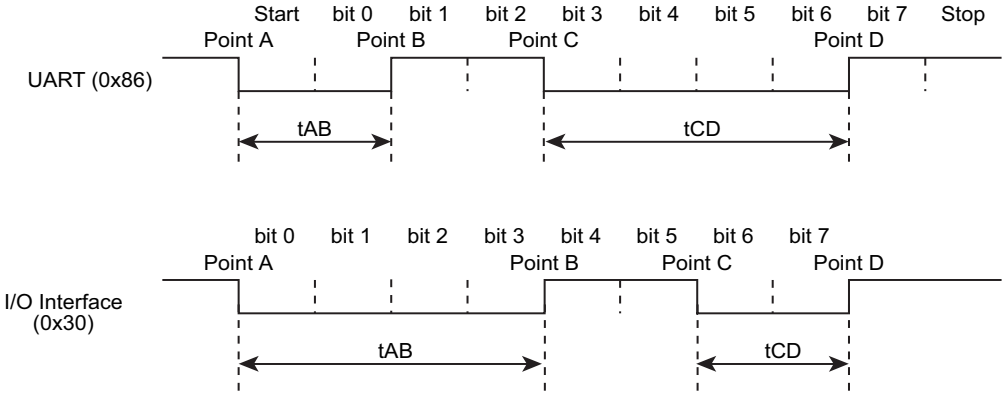


Figure 19-4 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Figure 19-5 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts when each time the logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals tend to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode may have this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 of the desired baud rate.

The flowchart in Figure 19-5 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Note that if the baud rate is too high or the timer operating frequency is too low, each timer value becomes small. It causes an unintentional behavior of the controller. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

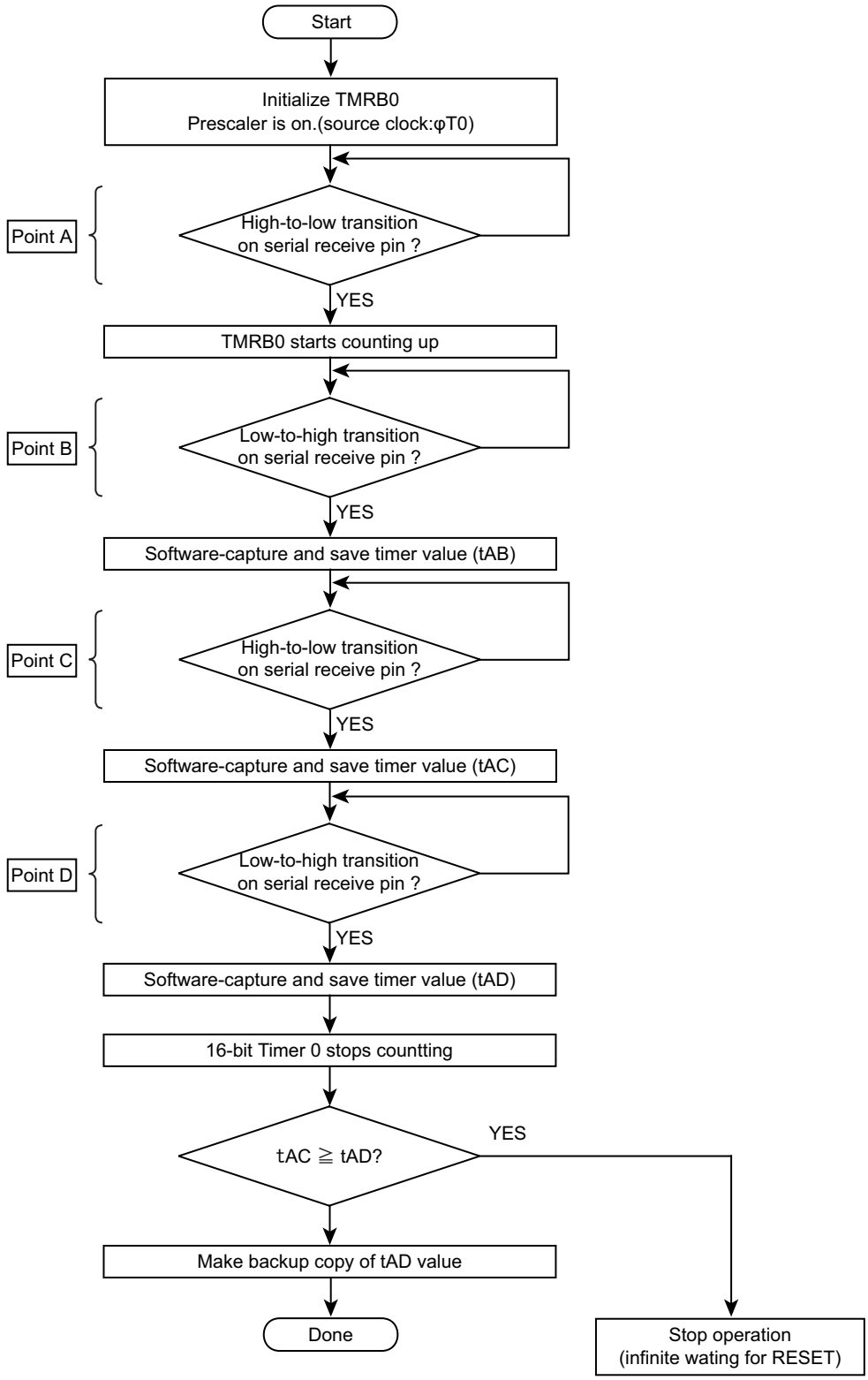


Figure 19-5 Serial Operation Mode Byte Reception Flowchart

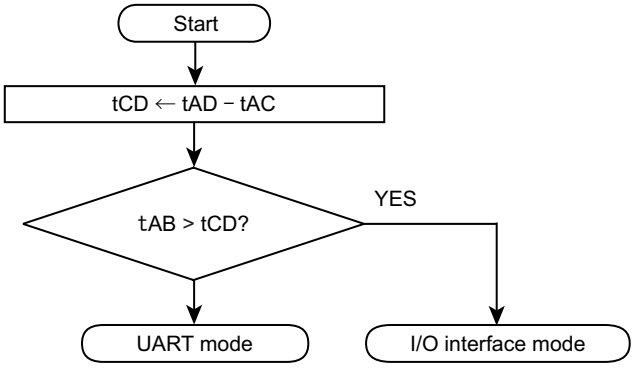


Figure 19-6 Serial Operation Mode Determination Flowchart

19.2.10.5Password

The RAM Transfer command (0x10) causes the boot program to perform password verification. Following an echo-back of the command code, the boot program verifies the contents of the 12-byte password area within the flash memory. The following table shows the password area of each product.

Product name	Area
TMPM376FDDFG/ FDFG	0x3F87_FFF4 to 0x3F87_FFFF

Note: **If a password is set to 0xFF (erased data area), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.**

If all these address locations contain the same bytes of data other than 0xFF, a password area error occurs as shown in Figure 19-7. 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.

Receiving data (5th to 16th bytes) from the controller is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

The password verification is performed even if the security function is enabled.

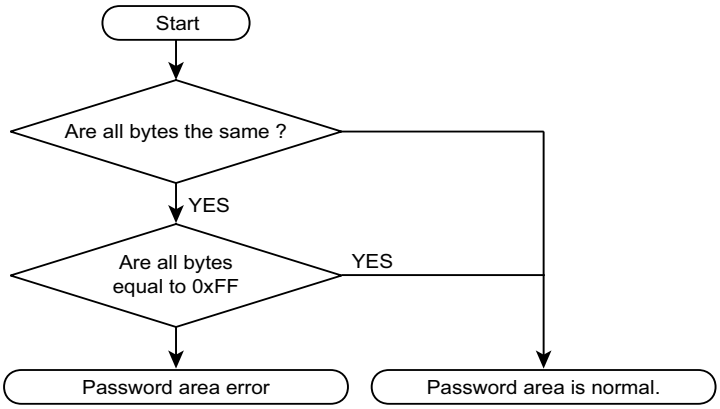


Figure 19-7 Password Area Verification Flowchart

19.2.10.6Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together with ignoring the carries and calculating the 8-bit two's complement by using lower 8 bits. The controller must perform the same checksum operation in transmitting checksum bytes.

Example) To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together

0xE5 + 0xF6 = 0x1DB

Calculate the two's complement by using lower 8 bits, and that is the checksum byte. Then send 0x25 to the controller.

0 - 0xDB = 0x25

19.2.11 General Boot Program Flowchart

Figure 19-8 shows an overall flowchart of the boot program.

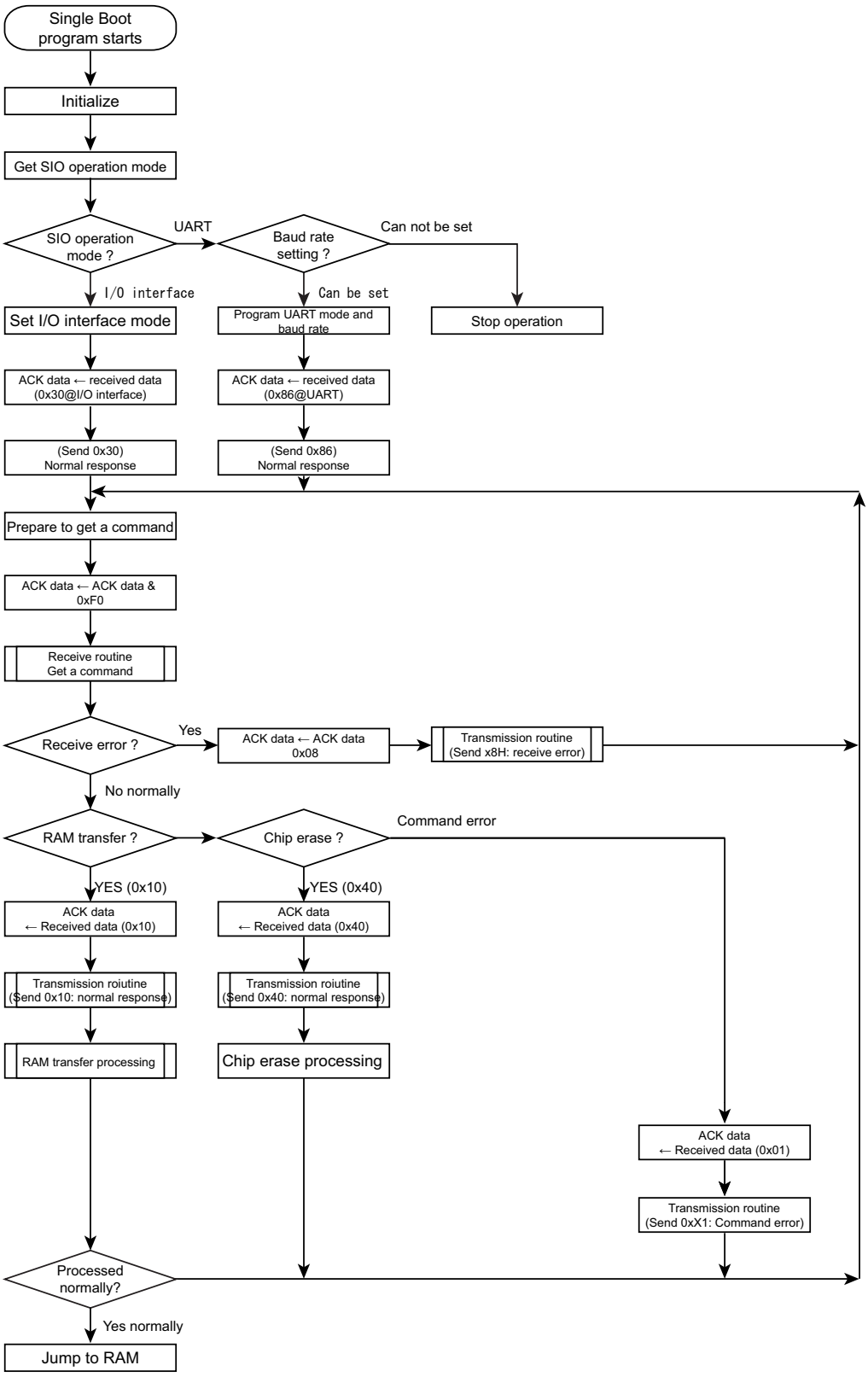


Figure 19-8 Overall Boot Program Flowchart

19.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM after shifting to the user boot mode.

19.3.1 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 19-12 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erase the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	The write or erase operation can be individually inhibited for each block.

19.3.1.1 Block Configuration

(1) TMPM376FDDFG / FDFG

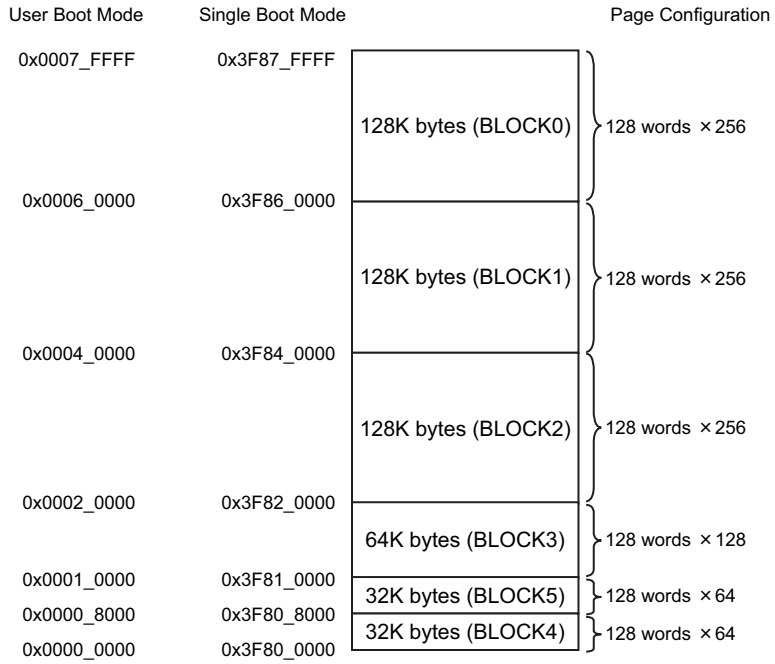


Figure 19-9 Block Configuration of Flash Memory (TMPM376FDDFG / FDFG)

19.3.1.2 Basic Operation

This flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exception other than reset and debug exceptions while a debug port is connected. Any exception generation cannot set the device to the read mode except when a hardware reset is generated.

(1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- Read / reset command and Read command (software reset)

  When ID-Read command is used, the reading operation is terminated instead of automatically returning to the read mode. In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used. The Read command is used to return to the read mode after executing 32-bit data transfer command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.
- With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.

(2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read.

While commands are generally comprised of several bus cycles and the operation applying to the 32-bit (word) data transmission command to the flash memory is called "bus write cycle". The bus write cycles have a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of command write is operated in accordance with a predefined specific order. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

Note 1: **Command sequences are executed from outside the flash memory area.**

Note 2: **Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, do not generate any interrupt (except debug exceptions when a debug port is connected). If such an operation is made, it may result in an unexpected read access to the flash memory, and the command sequencer may not be able to correctly recognize the command. While it may cause an abnormal termination of the command sequence, it also may cause an incorrect recognition of the command.**

Note 3: **For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle where FCFLCS <RDY / BSY> is set to "1". It is recommended to subsequently execute a Read command.**

Note 4: **Upon issuing a command, if any address or data is incorrectly written, be sure to perform a software reset to return to the read mode again.**

19.3.1.3 Reset (Hardware reset)

A hardware reset is used to cancel the operational mode set by the command write operation when forcibly terminated during auto programming/erasing or abnormal termination in the automatic operation.

The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the $\overline{\text{RESET}}$ input pin of this device is set to VIL or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section "19.2.1 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.

19.3.1.4 Commands

(1) Automatic Page Program

Writing to a flash memory device is to change "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For changing "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM376FDDFG/FDFG contains 128 words in a page. A 128 word block is defined by the same [31:9] address. It starts from the address [8:0] = 0x00 and ends at the address [8:0] = 0x1FF. This programming unit is hereafter referred to as a "page".

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by FCFLCS [0] <RDY/BSY>.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more. Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content may cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. After the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at one time). Be sure to use the 32-bit data transfer command in writing commands after the fourth bus cycle. At this time, any 32-bit data transfer commands shall not be placed across word boundary. After the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0". For example, if the top address of a page is not to be written, set the input data in the fourth bus write cycle to 0xFFFFFFFF as a command write.

Once the third bus cycle is executed, the automatic page programming is in operation. This condition can be checked by monitoring FCFLCS<RDY / BSY>. Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted. When a single page has been command written with normally terminating the automatic page writing process, FCFLCS<RDY / BSY> is set to "1" then it returns to the read mode.

When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FCFLCS<RDY/BSY>. If automatic programming has failed, the flash memory is locked in the current mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

Note: **Software reset becomes ineffective after the fourth bus write cycle of the automatic page programming command.**

(2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FCFLCS<RDY / BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected block cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the current mode and will not return to the read mode.

For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

(3) Automatic block erase (for each block)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FCFLCS <RDY / BSY>. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected block cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

(4) Automatic programming of protection bits (for each block)

This device is implemented with protection bits. This protection can be set for each block. See Table 19-16 for table of protection bit addresses. This device assigns 1 bit to 1 block as a protection bit. The applicable protection bit is specified by PBA in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FCFLCS <RDY/BSY>. Any new command sequence is not

accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, all FCFLCS <BLPRO> are set to "1" indicating that it is in the protected state. This disables subsequent writing and erasing of all blocks.

Note: **Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. FCFLCS <RDY/BSY> turns to "0" after entering the seventh bus write cycle.**

(5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on whether all <BLPRO> in the FCFLCS register are set to "1" or not, when FCSECBIT<FCSECBIT> is set to "1". Be sure to check the value of FCFLCS <BLPRO> before executing the automatic protection bit erase command. See Chapter "Protect/security function" for details.

- When all the FCFLCS <BLPRO> are set to "1" (all the protection bits are programmed):

  When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FCFLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FCFLCS will be set to "0x00000001". Since no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FCFLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as required.
- When FCFLCS <BLPRO> include "0" (not all the protection bits are programmed):

  If the automatic protection bit is cleared to "0", the protection condition is canceled. With this device, protection bits can be programmed to an individual block and performed bit-erase operation in the four bits unit as shown in Table 19-16. The target bits are specified in the seventh bus write cycle.The protection status of each block can be checked by FCFLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FCFLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FCFLCS <BLPRO> selected for erasure are set to "0".

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

Note: **The FCFLCS <RDY / BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.**

(6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). After the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repeatedly executed. For returning to the read mode, use the Read/reset command or hardware reset command.

19.3.1.5 Flash control / status register

Base Address = 0x41FF_F000

Register name		Address (Base+)
Reserved	-	0x0000
Reserved	-	0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024 to 0x0FFF

Note: **Do not access to the reserved address.**

(1) FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY_BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31 to 22	–	R	Read as 0.
21 to 16	BLPRO5 to BLPRO0	R	Protection for Block 5 to 0 0: disabled 1: enabled Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1", it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
15 to 1	–	R	Read as 0.
0	RDY/BSY	R	Ready / Busy (Note 1) 0:Auto operating 1:Auto operation terminated. Ready/Busy flag bit The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1".

Note 1: This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 μs regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

Note 2: The value varies depending on protection applied.

(2) FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	–	R	Read as 0.
0	SECBIT	R/W	Security bits 0:disabled 1:enabled

Note: **This register is initialized by cold reset.**

19.3.1.6 List of Command Sequences

Table 19-13 shows the address and the data of each command of flash memory.

Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus-cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by 32-bit (word) data transfer commands. (In the following table, only lower 8 bits data are shown.)

See Table 19-14 for the detail of the address bit configuration. Use a value of "Addr." in the Table 19-13 for the address [15:8] of the normal command in the Table 19-14.

Note: **Always set "0" to the address bits [1:0] in the entire bus cycle.**

Table 19-13 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read / Reset	0x54XX	0xAAXX	0x54XX	RA	–	–	–
	0xAA	0x55	0xF0	RD	–	–	–
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page pro-gramming	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 program-ming	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address
  PD: Program data (32 bit data)

  After fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

19.3.2 Address bit configuration for bus write cycles

Table 19-14 is used in conjunction with "Table 19-13 Flash Memory Access from the Internal CPU".

Address setting can be performed according to the normal bus write cycle address configuration from the first bus cycle. "0" is recommended" in the Table 19-14 Address Bit Configuration for Bus Write Cycles can be changed as necessary.

Address	Addr [31:19]	Addr [18]	Addr [17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10]	Addr [9]	Addr [8]	Addr [7:0]
Normal commands	Normal bus write cycle address configuration										
	Flash area	"0" is recommended.			Command						Addr[1:0]="0" (fixed) Others:0 (recommended)
ID-READ	IA: ID address (Set the fourth bus write cycle address for ID-Read operation)										
	Flash area	"0" is recommended.			ID address		Addr[1:0]="0" (fixed), Others:0 (recommended)				
Block erase	BA: Block address (Set the sixth bus write cycle address for block erase operation)										
	Block selection (Table 19-14)					Addr[1:0]="0" (fixed), Others:0 (recommended)					
Auto page program-ming	PA: Program page address (Set the fourth bus write cycle address for page programming operation)										
	Page selection										Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit pro-gramming	PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming)										
	Flash area	Protection bit selec-tion (Table 19-15)			Fixed to "0".			Protect bit selection (Table 19-15)			Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit erase	PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)										
	Flash area	Protection bit selection (Table 19-16)			Fixed to "0".						Addr[1:0]="0" (fixed) Others:0 (recommended)

As block address, specify any address in the block to be erased.

Refer to 19.3.1.1 for Block Configuration.

Table 19-14 Block Address Table

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
4	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32
5	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
3	0x0001_0000 to 0x0001_FFFF	0x3F81_0000 to 0x3F81_FFFF	64
2	0x0002_0000 to 0x0003_FFFF	0x3F82_0000 o 0x3F83_FFFF	128
1	0x0004_0000 to 0x0005_FFFF	0x3F84_0000 to 0x3F85_FFFF	128
0	0x0006_0000 to 0x0007_FFFF	0x3F86_0000 to 0x3F87_FFFF	128

Note: **As for the addresses from the first to the fifth bus cycles, specify the upper addresses of the blocks to be erased.**

Table 19-15 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 [9]
Block0	<BLPRO[0]>	0	0	1	Fixed to "0".	0	0	"0" is recommended.
Block1	<BLPRO[1]>	0	0	1		0	1	
Block2	<BLPRO[2]>	0	0	1		1	0	
Block3	<BLPRO[3]>	0	0	1		1	1	
Block4	<BLPRO[4]>	0	1	0		0	0	
Block5	<BLPRO[5]>	0	1	0		0	1	

Table 19-16 Protection Bit Erase Address Table

Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address [18]	Address [17]
Block0 to 3	<BLPRO[3:0]>	0	0

Note: **The protection bit erase command cannot erase by individual block.**

Table 19-17 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following 32-bit data transfer command (ID)

IA[15:14]	ID[7:0]	Code
0y00	0x98	Manufacturer code
0y01	0x5A	Device code
0y10	Reserved	–
0y11	0x12	Macro code

19.3.2.1 Flowchart

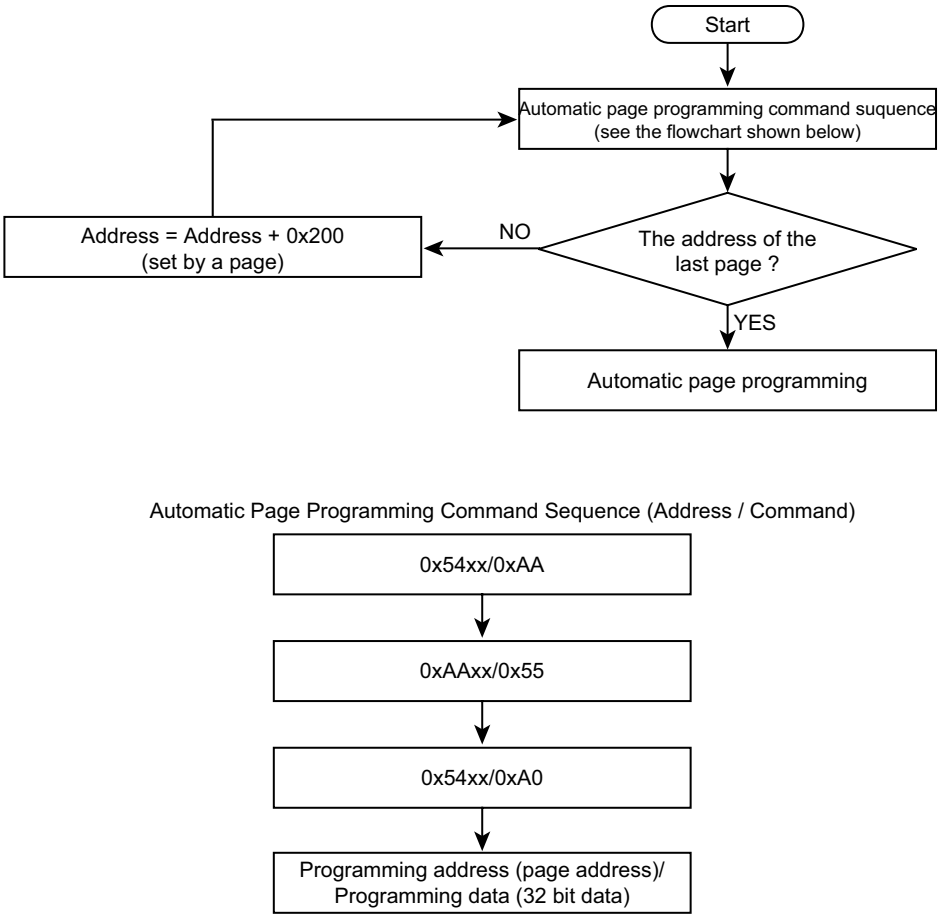


Figure 19-10 Automatic Programming

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

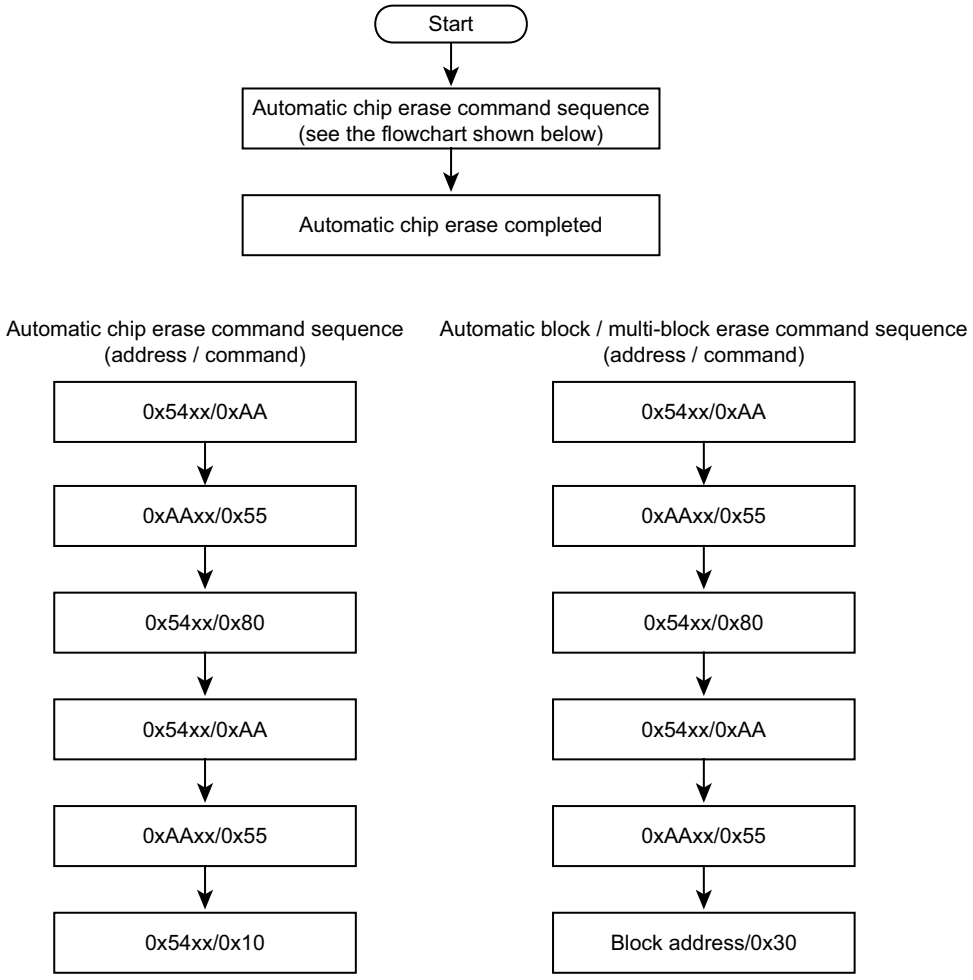


Figure 19-11 Automatic Erase

Note: **Command sequence is executed by 0x54xx or 0x55xx.**

20. ROM protection

20.1 Outline

The TMPM376FDDFG/FDFG offers two kinds of ROM protection/ security functions.

One is a write/ erase-protection function for the internal flash ROM data.

The other is a security function that restricts internal flash ROM data readout and debugging.

20.2 Future

20.2.1 Write/ erase-protection function

The write/ erase-protection function enables the internal flash to prohibit the writing and erasing operation for each block.

To activate the function, write "1" to the corresponding bits to a block to protect. Writing "0" to the bits cancels the protection.

The protection settings of the bits can be monitored by the FCFLCS <BLPRO[5:0]> bit. See the chapter "Flash" for programming details.

20.2.2 Security function

The security function restricts flash ROM data readout and debugging.

This function is available under the conditions shown below.

1. The FCSECBIT <SECBIT> bit is set to"1".
2. All the protection bits (the FCFLCS<BLPRO> bits) used for the write/erase-protection function are set to "1".

Note: The FCSECBIT <SECBIT> bit is set to "1" at a power-on reset right after power-on.

Table 20-1 shows details of the restrictions by the security function.

Table 20-1 Restrictions by the security function

Item	Details
1) ROM data readout	Data can be read from CPU.
2) Debug port	Communication of JTAG/SW and trace are prohibited
3) Command for flash memory	Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/ erase-protection erases all the protection bits.

20.3 Register

Base Address = 0x41FF_F000

Register name		Address(Base+)
Reserved	-	0x0000,0x0004
Security bit register	FCSECBIT	0x0010
Reserved	-	0x0014
Flash control register	FCFLCS	0x0020
Reserved	-	0x0024 to 0x0FFF

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

20.3.1 FCFLCS (Flash control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
After reset	0	0	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)	(Note2)
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY_BSY
After reset	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	Function
31-22	–	R	Read as 0.
21-16	BLPRO53 to BLPRO0	R	Protection for Block5 to 0 0: disabled 1: enabled Protection status bits Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.
17-1	–	R	Read as 0.
0	RDY_BSY	R	Ready/Busy (Note 1) 0: Auto operating 1:Auto operation terminated Ready/Busy flag bit The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Note 1: This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 ms regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

Note 2: The value varies depending on protection applied.

20.3.2 FCSECBIT(Security bit register)

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

Bit	Bit Symbol	Type	ã@î\
31-1	–	R	Read as 0.
0	SECBIT	R/W	Security bit 0: Disabled 1: Enabled

Note: This register is initialized by cold reset .

20.4 Writing and erasing

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.

20.4.1 Protection bits

Writing to the protection bits is done on block-by-block basis.

When the settings for all the blocks are "1", erasing must be done after setting the FCSECBIT <SECBIT> bit to "0". Setting "1" at that situation erases all the protection bits. To write and erase the protection bits, command sequence is used.

See the capter "Flash" for details

20.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to "1" at a power-on reset right after power-on.

The bit is rewritten by the following procedure.

1. Write the code 0xa74a9d23 to FCSECBIT register.
2. Write data within 16 clocks from the above.1.

Note: The above procedure is enabled only when using 32-bit data transfer command.

21. Debug Interface

21.1 Specification Overview

The TMPM376FDDFG/FDFG 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] to [1], SWV) via the on-chip Trace Port Interface Unit (TPIU).

21.2 Features of SWJ-DP

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

21.3 Features of ETM

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

21.4 Pin Functions

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

Table 21-1 SWJ-DP, ETM function

SWJ-DP Pin name	Name of port	JTAG debug function		SW debug	
		I/O	Description	I/O	Description
TMS/SWDIO	PB3	Input	JTAG Test Mode Selection	I/O	Serial Wire Data Input/Output
TCK/SWCLK	PB4	Input	JTAG Test Check	Input	Serial Wire Clock
TDO/SWV	PB5	Output	JTAG Test Data Output	(Input) (Note1)	(Serial Wire Viewer Output)
TDI	PB6	Input	JTAG Test Data Input	-	-
$\overline{\text{TRST}}$	PB7	Input	JTAG Test RESET	-	-
TRACECLK	PB0	Output	TRACE Clock Output		
TRACEDATA0	PB1	Output	TRACE DATA Output0		
TRACEDATA1	PB2	Output	TRACEDATA Output1		

Note: In case of enabling SWV function

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

Table 21-2 below summarizes the debug interface pin functions and related port settings after reset.

Table 21-2 Debug interface pins and port setting after reset

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

When using a low power consumption mode, take note of the following points.

Note 1: If PB3 and PB5 are configured as debug function pins, output continues to be enabled even in STOP mode regardless of the setting of the CGSTBYCR<DRVE>.

Note 2: If PB4 is configured as a debug function pin, it prevents a low power consumption mode from being fully effective. Configure PB4 to function as a general-purpose port if the debug function is not used.

21.5 Connection with a Debug Tool

21.5.1 How to connect

For how to connect a debug tool, refer to the method recommended by each manufacture.Debug interface pins have pull-up or pull-down register. When connect with pull-up or pull-down riggers, be sure their settings.

21.5.2 When use general purpose port

When debugging, do not change setting debug interface to general purpose port by program. Then, MCU will be unable to control signals received from the debugging tools and can not continue debugging. According to the usage of the debug interface pins, be sure their setting.

Table 21-3 Debug Interface

Usage	Using Debug Interface (O:Enable, -:Disable)							
	$\overline{\text{TRST}}$	TDI	TDO/ SWV	TCK/ SWCLK	TMS/ SWDIO	TRACE DATA1	TRACE DATA0	TRACE CLK
JTAG+SW (After RESET)	O	O	O	O	O	-	-	-
JTAG+SW (non $\overline{\text{TRST}}$)	-	O	O	O	O	-	-	-
JTAG+TRACE	O	O	O	O	O	O	O	O
SW	-	-	-	O	O	-	-	-
SW+SWV	-	-	O	O	O	-	-	-
Disable Debug function	-	-	-	-	-	-	-	-

21.6 Peripherals operation during HALT mode

When Break during debugging, Cortex-M3 CPU core going into HALT mode. Watch dog timer (WDT) is stopped counting automatically. And 16bit timer/counter can specify the status (continue operating or stop) in HALT mode. Other peripherals are continue operating.

22. Electrical Characteristics

22.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5	−0.3 to 6	V
		RVDD5	−0.3 to 6	
		AVDD5A/B	−0.3 to 6	
Capacitor voltage		VOUT15	−0.3 to 3	V
		VOUT3	−0.3 to 3.9	
Input voltage		V_{IN}	−0.3 to VDD + 0.3 (Note2)	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	50	
Power consumption (Ta = 85 °C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−55 to 125	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	−40 to 85	°C
	During Flash W/E		0 to 70	

Note 1: 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 2: **VDD** = **DVDD5** = **RVDD5** = **AVDD5A** = **AVDD5B**

22.2 DC Electrical Characteristics (1/2)

DVSS = AVSSA = AVSSB = 0V, Ta = –40 to 85 °C

Parameter		Symbol	Rating		Min.	Typ. (Note 1)	Max.	Unit
Supply voltage (Note 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f_{OSC} = 8 to 10 MHz fsys = 1 to 80 MHz		4.5	–	5.5	V
Supply voltage (during Flash W/E) (Note 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f_{OSC} = 8 to 10 MHz fsys = 1 to 80 MHz (Ta (°C) = 0 to 70)		4.5	–	5.5	V
Supply voltage (Power-on or Power-off) (Note 3)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f_{OSC} = 8 to 10 MHz fsys = 1 to 80 MHz		3.9	–	5.5	V
Low-level input voltage	Schmitt-Input	V_{IL1}	VDD = 4.5V to 5.5V (Note 4)		–0.3	–	0.25 VDD	V
Hight-level input voltage	Schmitt-Input	V_{IH1}	VDD = 4.5V to 5.5V (Note 4)		0.75VDD		VDD+0.3	V
Capacitance for VOUT15 and VOUT3 (Note 3)		C_{out}	RVDD5 = 4.5V to 5.5V VOUT15, VOUT3		3.3	–	4.7	μF
Low-level output voltage		V_{OL}	I_{OL} = 1.6 mA	VDD ≥ 4.5V (Note 4)	–	–	0.4	V
High-level output voltage		V_{OH}	I_{OH} = –1.6 mA	VDD ≥ 4.5V (Note 4)	4.1	–	–	V
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ VDD (Note 4)		–	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ VDD -0.2 (Note 4)		–	0.05	±10	
Pull-up resister at Reset		R_{RST}	4.5 ≤ VDD ≤ 5.5 (Note 4)		–	50	150	kΩ
Programmable pull-up/pull-down resistor		P_{KH}	4.5 ≤ VDD ≤ 5.5 (Note 4)		–	50	150	kΩ
Schmitt-Triggered port		V_{TH}	4.5 ≤ VDD ≤ 5.5 (Note 4)		0.3	0.6	–	μF
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz		–	–	10	pF

Note 1: Ta = 25 °C, DVDD5 = AVDD5A = AVDD5B = RVDD5 = 5V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD5, DVDD5A, DVDD5B and RVDD5 .

Note 3: It is a voltage range in the case of Power-on or Power-off (when VLTD disabled). In the range whose Power-line is 3.9V ≤ VDD < 4,5V, does not guarantee a 12-bit A/D converter and AC electrical Characteristics. Please refer to a figure (Powe on Sequence (Using Power On Reset only)) for details.

Note 4: VOUT15 and VOUT3 pin should be connected to GND via same value of capacitance. The IC outside can not have the power supply from VOUT15 and VOUT3.

Note 5: VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B

22.3 DC Electrical Characteristics (2/2)

DVDD5 =RVDD5 = AVDD5A = AVDD5B = 4.5 V to 5.5 V, Ta = −40 to 85 °C

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL (Note 2) Gear 1/1	I_{DD}	fsys = 80 MHz	−	70	80	mA
IDLE (Note 4) Gear 1/1			−	21	30	
STOP		−	−	7	11	mA

Note 1: Ta=25°C, DVDD5 = AVDD5A = AVDD5B = RVDD5 = 5V, unless otherwise.

Note 2: I_{DD} NORMAL:
All functions operates excluding A/D.

Note 3: A/D reference voltage supply can not go into off state.

Note 4: I_{DD} IDLE :
All peripheral functions stopped.

22.4 12-bit ADC Electrical Characteristics

DVDD5 = RVDD5 = AVDD5A / VREFHA = AVDD5B / VREFHB = 4.5 V to 5.5 V
DVSS = AVSSA / VREFLA = AVSSB / VREFLB = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Rating	Min.	Typ.	Max	Unit
Analog reference voltage (+)		VREFHA VREFHB	−	−	AVDD	−	V
Analog input voltage		VAIN	−	AVSS	−	AVDD	V
Analog supply current (Note 1), (Note 2)		IREF	DVSS = AVSS	−	3.5	5.0	mA
Supply current (Note 1)	A/D conversion	−	Except IREF	−	−	6.0	mA
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 μF Conversion time ≥ 2 μs	−	−	± 6	LSB
DNL error				−	−	± 5	
Offset error				−	−	± 5	
Full-scale error				−	−	± 5	
Total error				−	−	−10 to +5	

Note 1: Current for one unit of ADC.
Note 2: A/D reference voltage supply can not go into off state.
Note 3: 1LSB = (AVDD − AVSS)/4096 [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.

22.5 AC Electrical Characteristics

22.5.1 AC measurement condition

AC measurement condition

- Output levels: High = 0.8 × VDD / Low = 0.2 × VDD
- Input levels: Refer to low-level input voltage and high-level input voltage in DC Electrical Characteristics.
- Load capacity : CL = 30pF

Note: VDD = DVDD5 = AVDD5A = AVDD5B

22.5.2 Serial Channel Timing (SIO/UART)

22.5.2.1 I/O Interface mode (VDD=4.5V to 5.5V)

In the table below, the letter x represents the period of the system clock (fsys). It varies depending on the programming of the clock gear function.

(1) SCLK input mode (Ta = –40 to 85°C)

[Input]

Parameter	Symbol	Equation		80 MHz		Unit
		Min.	Max	Min.	Max	
SCLK Clock High width (input)	t_{SCH}	4x	–	50	–	ns
SCLK Clock Low width (input)	t_{SCL}	4x	–	50	–	
SCLK cycle	t_{SCY}	8x	–	100	–	
Input Data valid SCLK rise or fall (Note1)	t_{SRD}	30	–	30	–	
InputData hold or fall after SCLK rising (Note 1)	t_{HSR}	x + 30	–	42.5	–	

[Output]

Parameter	Symbol	Equation		80 MHz		Unit
		Min.	Max	Min.	Max	
SCLK Clock High width (input)	t_{SCH}	4x	–	82.5 (Note 3)	–	ns
SCLK Clock Low width (input)	t_{SCL}	4x	–	82.5 (Note 3)	–	
SCLK cycle	t_{SCY}	8x	–	165	–	
OutputData to SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 4x - 45$ (Note2)	–	0 (Note 2)	–	
InputData hold or fall after SCLK rising (Note 1)	t_{OHS}	$t_{SCY}/2$	–	82.5	–	

Note 1: SCLK rise or fall:
Measured relative to the programmed active edge of SCLK.

Note 2: A calculated value should use it the SCLK cycle of the range which is not subtracted.

Note 3: t_{OSS} shows the minimum which is not subtracted.

(2) SCLK Output mode (Ta = −40 to 85°C)

[Output]

Parameter		Symbol	Equation		80 MHz		Unit
			Min.	Max	Min.	Max	
SCK cycle (programmable)		t_{SCY}	4x	−	60	−	ns
Output Data	← SCK rise	t_{OSS}	$t_{SCY}/2 - 30$ (Note1)	−	0 (Note2)	−	
SCK rise	→ Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$ (Note1)	−	0 (Note2)	−	
Valid Data input	← SCK rise	t_{SRD}	45	−	45	−	
SCK rise	→ Input Data hold	t_{HSR}	0	−	0	−	

Note 1: A calculated value should use it the SCLK cycle of the range which is not subtracted.
Note 2: t_{OSS} shows the minimum which is not subtracted.

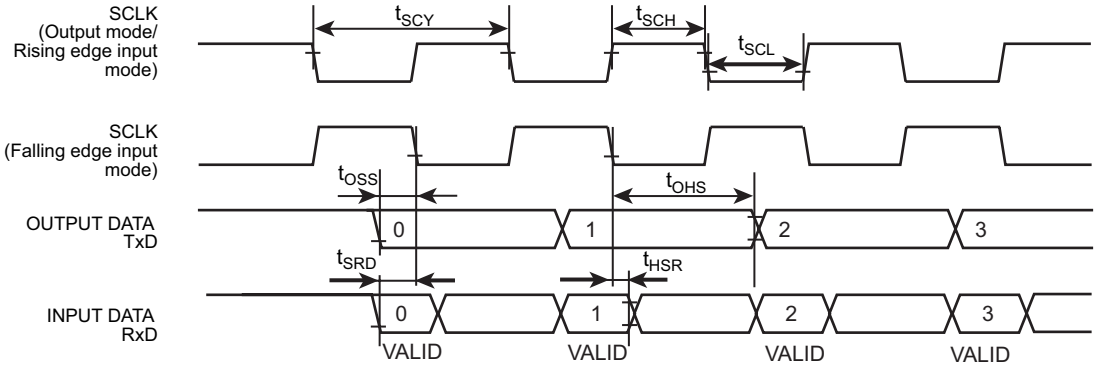


Figure 22-1 Serial channel timing(SIO/UART)

22.5.3 Serial Bus Interface (I2C/SIO)

22.5.3.1 I2C Mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBIxCR.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min.	Max	Min.	Max	Min.	Max	
SCL clock frequency	t_{SCL}	0	–	0	100	0	400	kHz
Hold time for START condition	$t_{HD;\ STA}$	–	–	4.0	–	0.6	–	µs
SCL Low width (Input) (Note 1)	t_{LOW}	–	–	4.7	–	1.3	–	µs
SCL High width (Input) (Note 2)	t_{HIGH}	–	–	4.0	–	0.6	–	µs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note 5)	–	4.7	–	0.6	–	µs
Data hold time (Input) (Note 3) (Note 4)	$t_{HD;\ DAT}$	–	–	0.0	–	0.0	–	µs
Data setup time	$t_{SU;\ DAT}$	–	–	250	–	100	–	ns
Setup time for a STOP condition	$t_{SU;\ STO}$	–	–	4.0	–	0.6	–	µs
Bus free time between stop condition and start condition	t_{BUF}	(Note 5)	–	4.7	–	1.3	–	µs

Note 1: SCL clock Low width (output) = $(2^{n-1} + 58)/x$

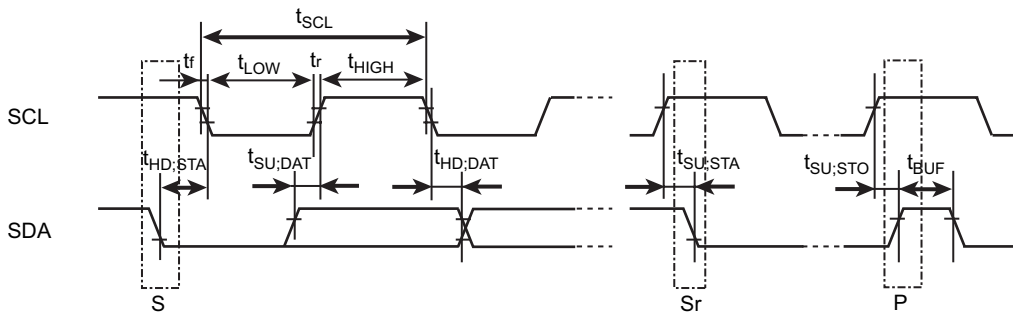
Note 2: SCL clock High width (output) = $(2^{n-1} + 14)/x$
On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400khz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

Note 3: The output data hold time is equal to 4x of internal SCL.

Note 4: The Philips I2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL. However, this SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software -dependent

Note 6: The Philips I2C-bus specification instructs that if the power supply to a Fast-mode device is switched off, the SDA and SCL I/O pins must be floating so that they don't obstruct the bus lines. However, this SBI does not satisfy this requirement.



S: Start condition
Sr: Re-start condition
P: Stop condition

Figure 22-2 Serial Bus timing (I2C/SIO)

22.5.3.2 Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsy-scycle time. It varies depending on the programming of the clock gear function.

(1) SCK Input Mode (SCK signal with a 50% duty cycle)

[Input data]

Parameter		Symbol	Equation		80 MHz		Unit
			Min.	Max	Min.	Max	
SCK Clock High width (input)		t_{SCH}	4x	–	82.5	–	ns
SCK Clock Low width (input)		t_{SCL}	4x	–	82.5	–	
SCK cycle		t_{SCY}	$8xt_{SCH} + t_{SCL}$	–	165	–	
Valid Data input	←SCK rise	t_{SRD}	30 – x		17.5	–	
SCK rise	→Input Data hold	t_{HSR}	30	–	30	–	

[Output data]

Parameter		Symbol	Equation		80 MHz		Unit
			Min.	Max	Min.	Max	
SCK Clock High width (input)		t_{SCH}	4x	–	5082.5	–	ns
SCK Clock Low width (input)		t_{SCL}	4x	–	82.5	–	
SCK cycle		t_{SCY}	$8xt_{SCH} + t_{SCL}$	–	165	–	
Output Data	←SCK rise	t_{OSS}	$t_{SCY}/2 – 3x – 45$	–	0 (Note)	–	
SCK rise	→Output Data hold	t_{OHS}	$t_{SCY}/2 + x$		95	–	

Note: Keep this value positive by adjusting SCK cycle.

(2) SCK Output Mode (SCK signal with a 50% duty cycle)

Parameter		Symbol	Equation		80 MHz		Unit
			Min.	Max	Min.	Max	
SCK cycle (programmable)		t_{SCY}	16x	–	200	–	ns
Output Data	←SCK rise	t_{OSS}	$t_{SCY}/2 – 20$	–	80	–	
SCK rise	→Output Data hold	t_{OHS}	$t_{SCY}/2 – 20$	–	80	–	
Valid Data input	←SCK rise	t_{SRD}	45	–	45	–	
SCK rise	→Input Data hold	t_{HSR}	0	–	0	–	

Note 1: SCK cycle after automatic wait becomes 14x.
Note 2: t_{OSS} after automatic wait may be $t_{SCY}/2$-x-20.

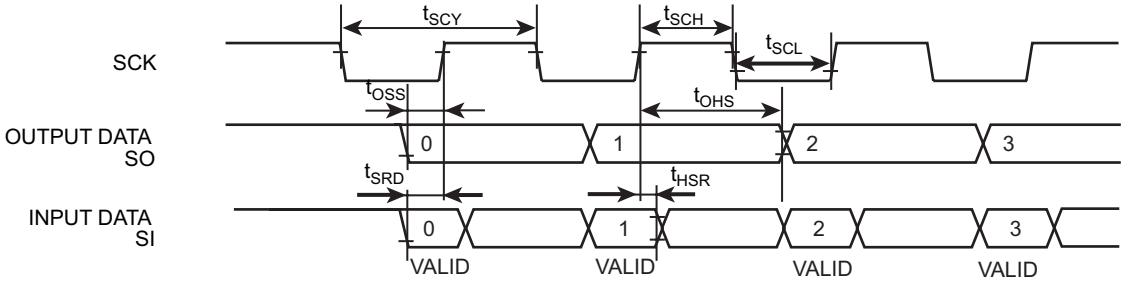


Figure 22-3 Serial Bus timing (SIO)

22.5.4 Event Counter

The character x shows the period of the clock for TMRB. The clock of TMRB is the same cycle as a system clock (fsys). It varies depending on the programming of the clock gear function.

Ta = –40 to 85°C (1 to 80MHz)

Parameter	Symbol	Equation		80 MHz		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

22.5.5 Capture

The character x shows the period of the clock for TMRB. The clock of TMRB is the same cycle as a system clock (fsys). It varies depending on the programming of the clock gear function.

Ta = –40 to 85°C (1 to 80MHz)

Parameter	Symbol	Equation		80 MHz		Unit
		Min.	Max	Min.	Max	
Low pulse width	t_{CPL}	2x + 100	–	125	–	ns
High pulse width	t_{CPH}	2x + 100	–	125	–	ns

22.5.6 External Interrupt

In the table below, the letter x represents the period of the system clock (fsys).

Ta= –40 to 85°C (1 to 80MHz)

1. Except STOP release interrupts

Parameter	Symbol	Equation		80 MHz		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

2. STOP Release Interrupts

Parameter	Symbol	Equation				Unit
		Min.	Max	Min.	Max	
Low pulse width for INT0 to F	t_{INTBL}	100	–	100	–	ns
High pulse width for INT0 to F	t_{INTBH}	100	–	100	–	ns

22.5.7 Debug Communication

22.5.7.1 AC measurement condition

- Output levels : High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Load capacitance : CL(TRACECLK) = 25pF, CL(TRACEDATA) = 20pF

22.5.7.2 SWD Interface

Parameter	Symbol	Min.	Max	Unit
CLK cycle	T_{dck}	100	–	ns
DATA hold after CLK rising	T_{d1}	4	–	
DATA valid after CLK rising	T_{d2}	–	37	
DATA valid to CLK rising	T_{ds}	20	–	
DATA hold after CLK falling	T_{dh}	15	–	

22.5.7.3 JTAG Interface

Parameter	Symbol	Min.	Max	Unit
CLK cycle	T_{dck}	100	–	ns
DATA hold after CLK falling	T_{d3}	4	–	
DATA valid after CLK falling	T_{d4}	–	37	
DATA valid to CLK rising	T_{ds}	20	–	
DATA hold after CLK rising	T_{dh}	15	–	

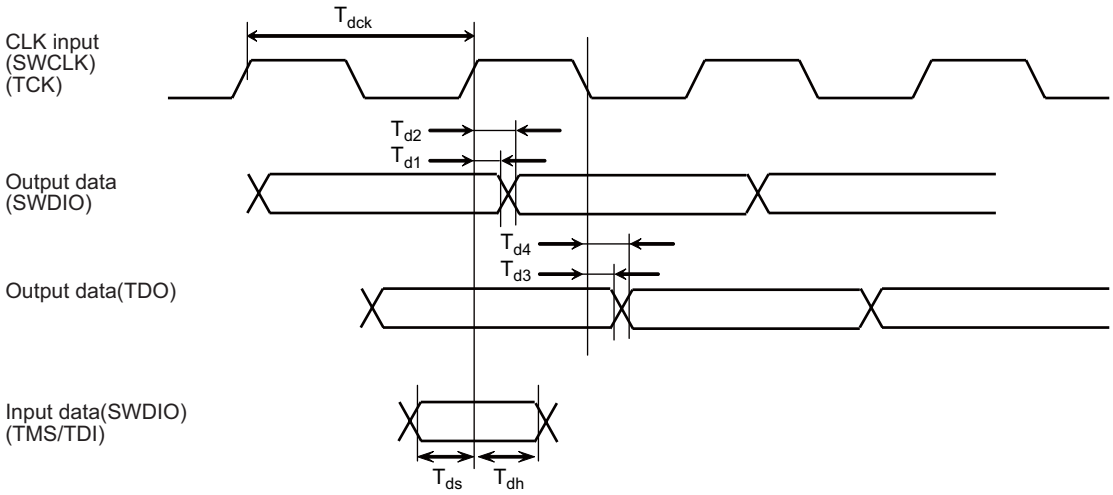


Figure 22-4 JTAG and SWD communication timing

22.5.8 TRACE Output

AC measurement condition

- Output levels : High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Load capacitance : CL(TRACECLK) = 25pF, CL(TRACEDATA) = 20pF

Parameter	Symbol	Min.	Max	Unit
TRACECLK cycle	t_{tclk}	25	–	ns
DATA valid after CLK rising	t_{setupr}	2	–	
DATA hold after CLK rising	t_{holdr}	1	–	
DATA valid after CLK falling	t_{setupf}	2	–	
DATA hold after CLK falling	t_{holdf}	1	–	

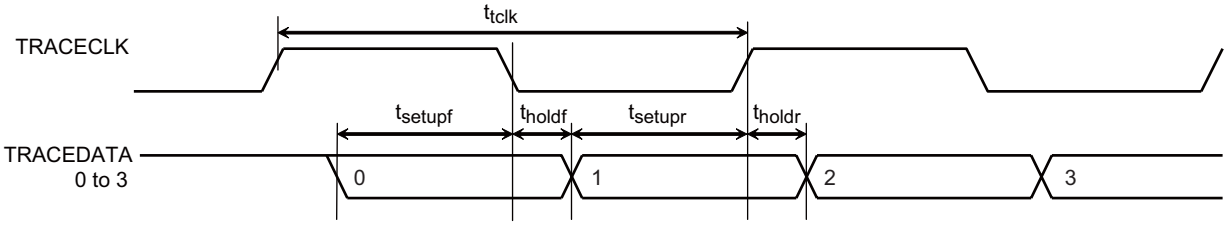


Figure 22-5 TRACE communication timing

22.5.9 Flash Characteristics

Parameter	Rating	Min.	Typ.	Max	Unit
Guarantee on Flash-memory rewriting	Ta = 0 to 70°C DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5 to 5.5V	–	–	100	times

22.5.10Internal Oscillator

Parameter	Symbol	Rating	Min.	Typ.	Max	Unit
Oscillation frequency	fosc2	Ta = -40 to 85°C	9.0	9.5	10	MHz

22.5.11External Oscillator

Parameter	Symbol	Rating	Min.	Typ.	Max	Unit
High frequency Oscillation	fosc1	Ta = -40 to 85°C	8	–	10	MHz

22.6 Oscillation Circuit

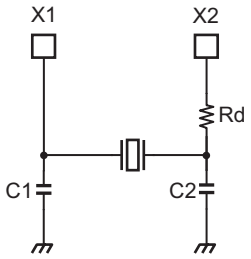


Figure 22-6 High-frequency oscillation connection

Note 1: The load value of the oscillator is the sum of loads (C1 and C2) and the floating load of the actual assembled board. There is a possibility of operating error when using C1 and C2 values in the table below. When designing the board, design the minimum length pattern around the oscillator. We also recommend that oscillator evaluation be carried out using the actual board.

Note 2: Do not be driven X1/X2 by external driver.

The TX03 has been evaluated by the oscillator vender below. Use this information when selecting external parts.

22.6.1 Recommended ceramic oscillator

The TX03 recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.

Please refer to the following URL for details.

http://www.murata.co.jp

22.7 Notes on the power on

Note for usage of Port L(PL0 and PL1 pin) when power on.

When power on, until VDD reach operation voltage and passed 200μs, port L(PL0 and PL1 pin) 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: VDD = DVDD5 = RVDD5 = DVDD5A = DVDD5B

22.7.1 Using Power On Reset only

Note 1: When you start a power supply using built-in power on reset, DVDD5 and RVDD5 terminal should start a power supply to reach the recommendation operation voltage range (3.9 to 5.5V) within 3 ms.

Note 2: Please choose arbitrary disregard levels after the start of a microcomputer of operation in a voltage detector circuit (VLTD), and enable operation.

Symbol	Rating	Min.	Typ.	Max	Unit
t_{PWUP}	Warming-up time after reset released	-	-	3.7	ms
t_{DVDD}	Rising time of power line	-	-	3	
V_{VLTD}	Detection voltage of a voltage detector circuit (In the case of VDCR<VDLVL[1:0]> ="01")	3.9	4.1	4.3	V
V_{PORH}	Power-on Reset releasing voltage	2.8	3	3.2	
V_{PORL}	Power-on Reset detection voltage	2.6	2.8	3.0	

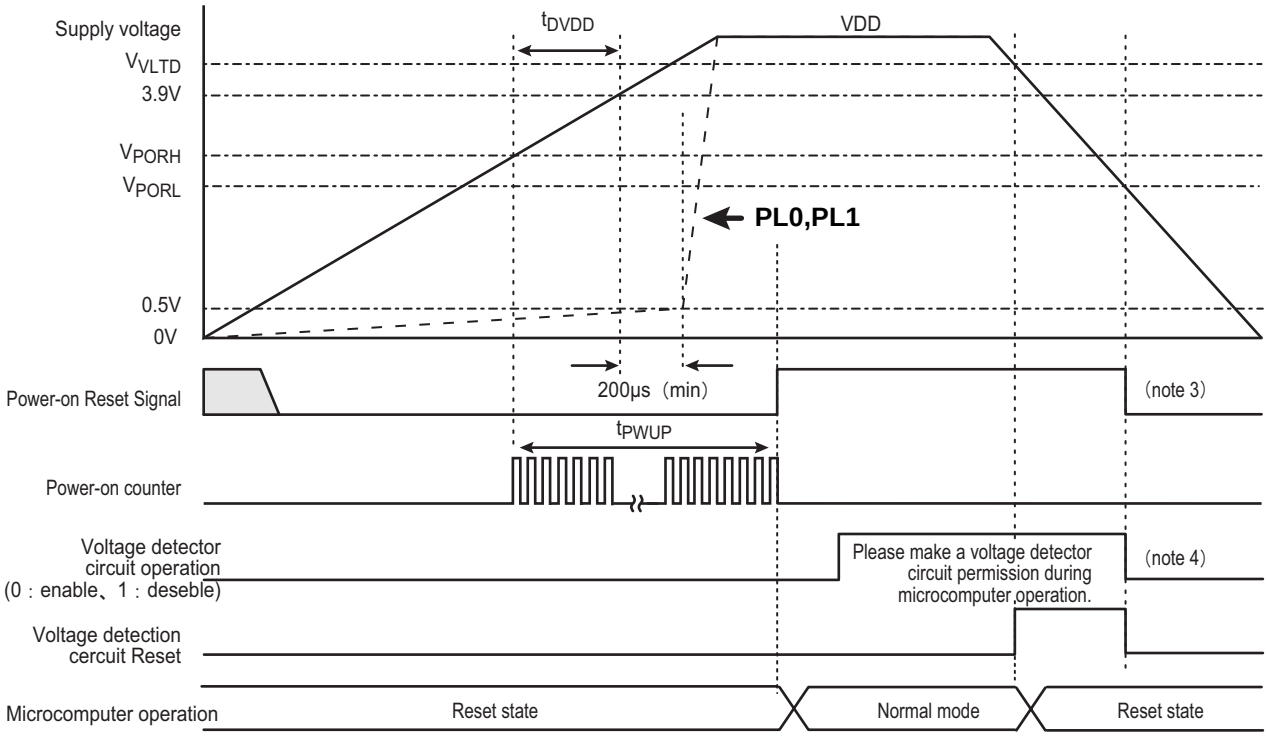


Figure 22-7 Powe on Sequence (Using Power On Reset only)

Note 1: VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B

Note 2: Since power-on-reset release voltage (V_{PORH}) and power-on-reset detection voltage(V_{PORL}) are changed relatively, detection voltage does not reverse them.

Note 3: If power supply voltage becomes V_{PORL} or less, power on reset will start.

Note 4: A voltage detector circuit (VLTD) is initialized (= VLTD is disable) by power-on-reset generating.

22.7.2 Using External Reset

22.7.2.1 IN case of the time of external reset shorter then POR

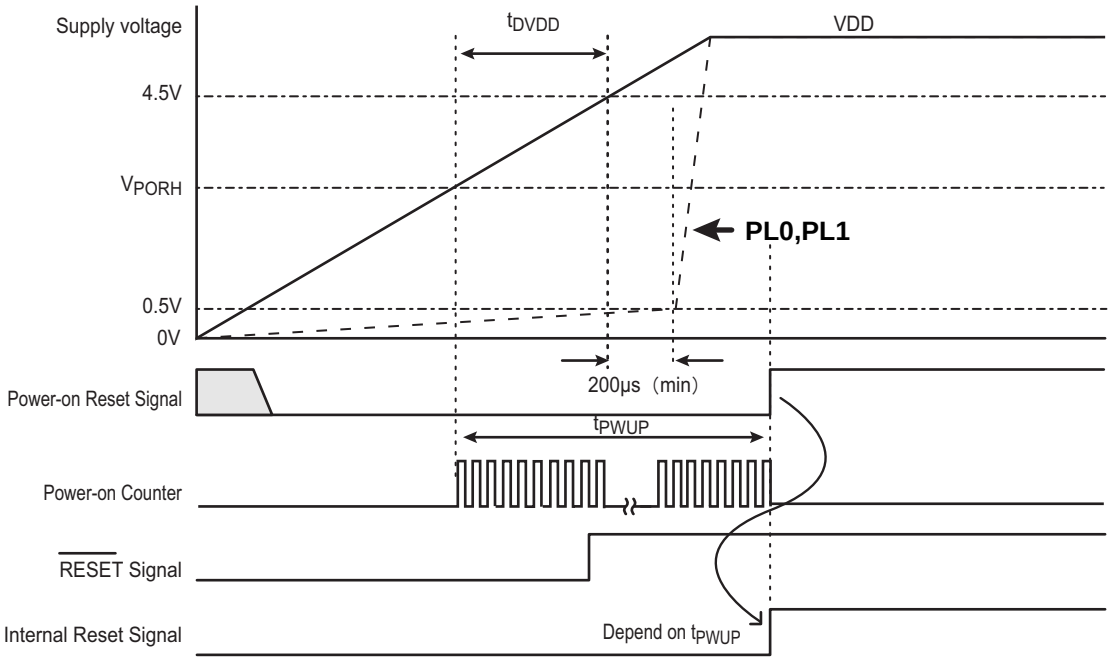


Figure 22-8 Power on Sequence (Using POR and External reset) (1)

Note: VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B

22.7.2.2 IN case of the time of external reset longer then t_{PWUP}

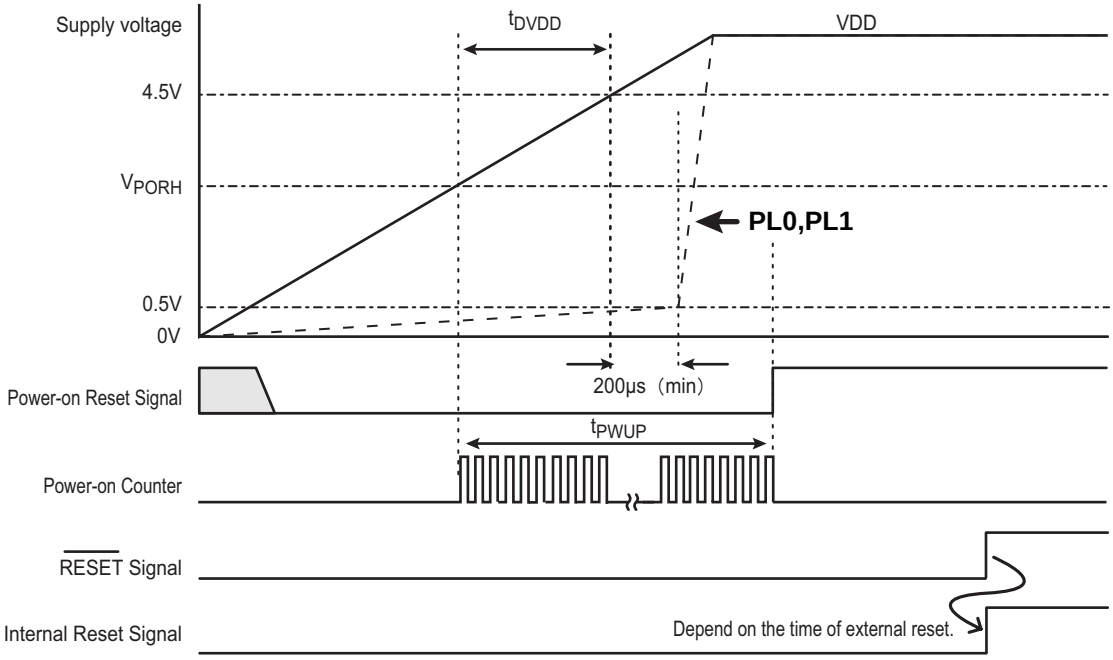


Figure 22-9 Power on Sequence (Using POR and External reset) (2)

Note 1: VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B

22.7.2.3 IN case of the rising time of power line longer then t_{PWUP}

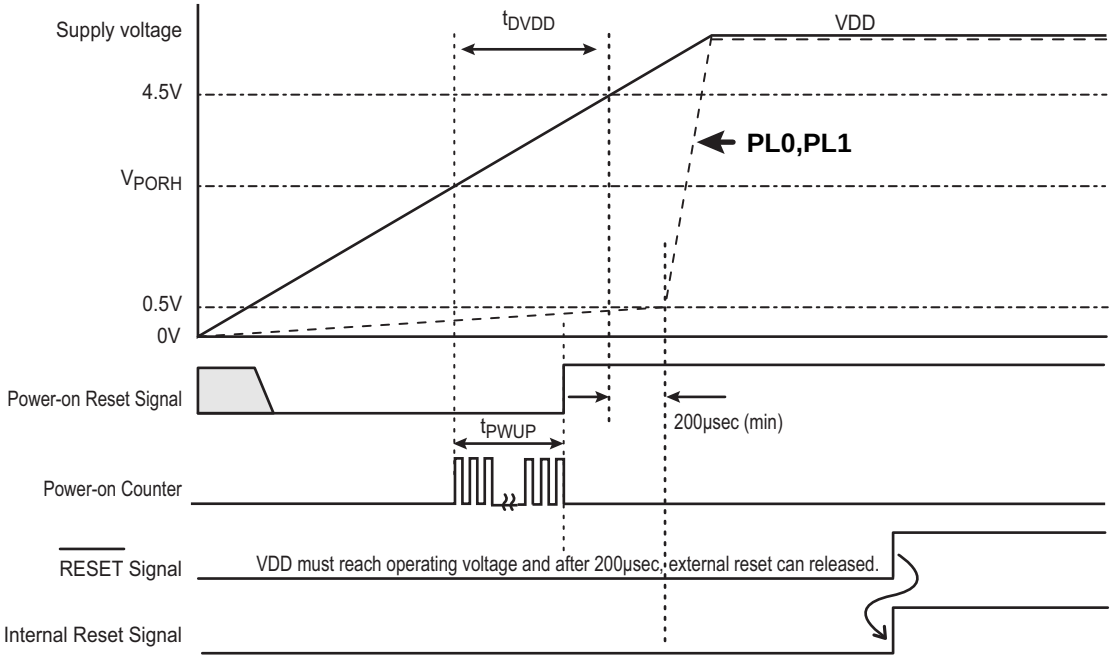


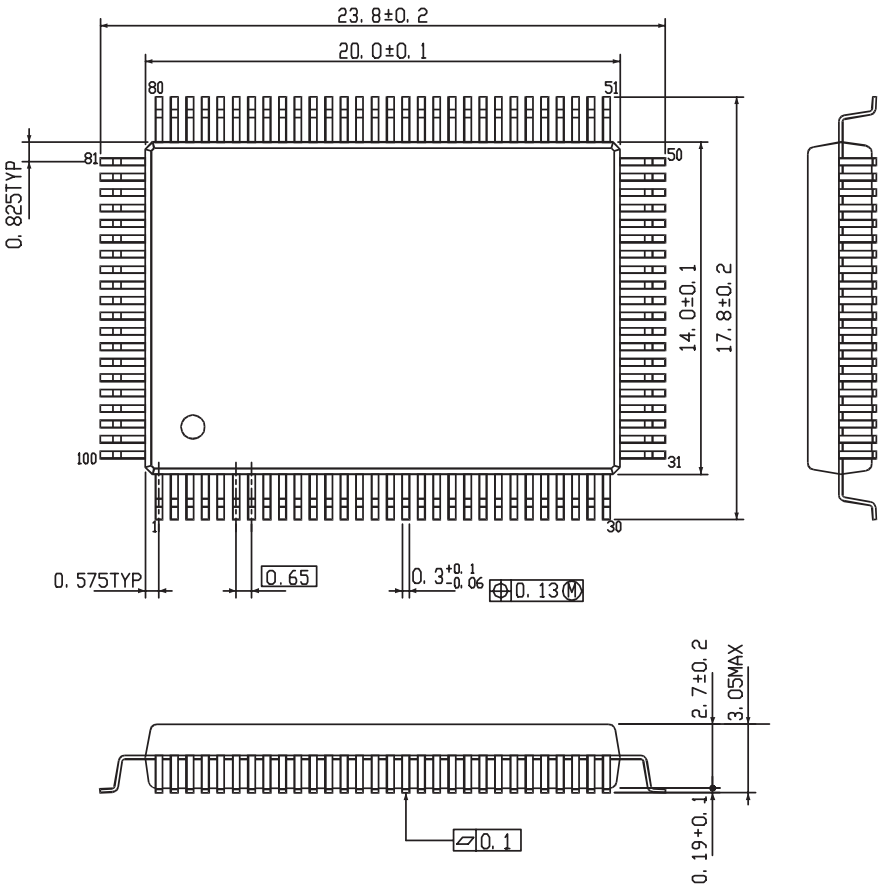
Figure 22-10 Power on Sequence ($t_{DVDD} > t_{PWUP}$)

Note 1: VDD =DVDD5 = RVDD5 = AVDD5A = AVDD5B
Note 2: In this case, must be reset from $\overline{\text{RESET}}$ pin.

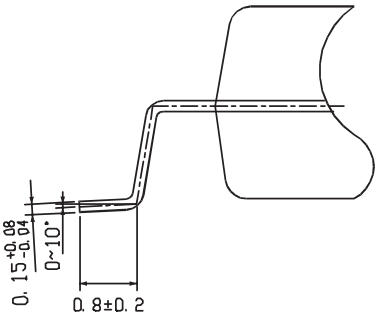
23. Package Dimensions

23.1 Type: P-QFP100-1420-0.65Q

Dimensions Unit: mm



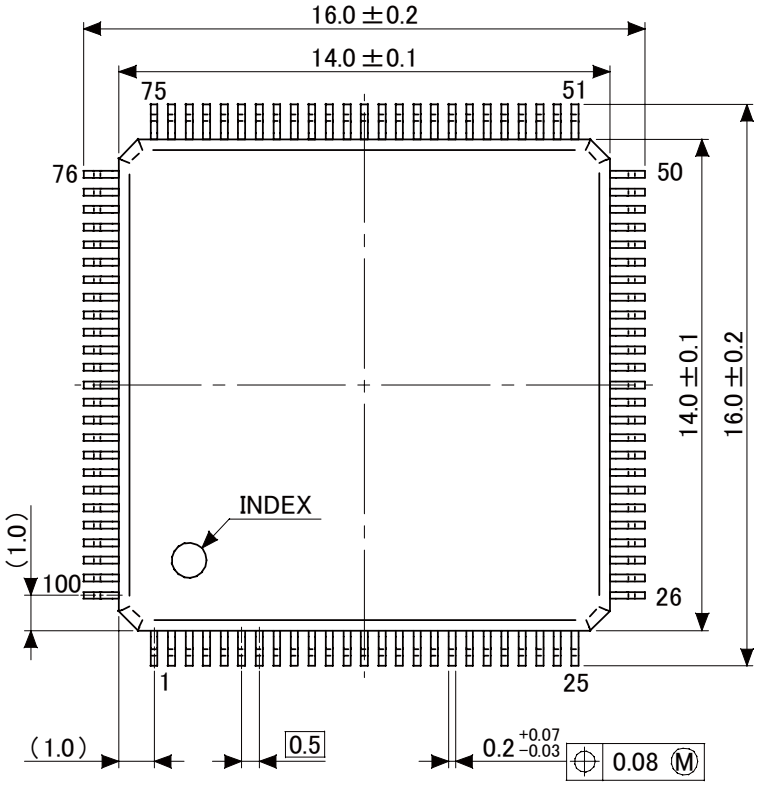
Pin detail

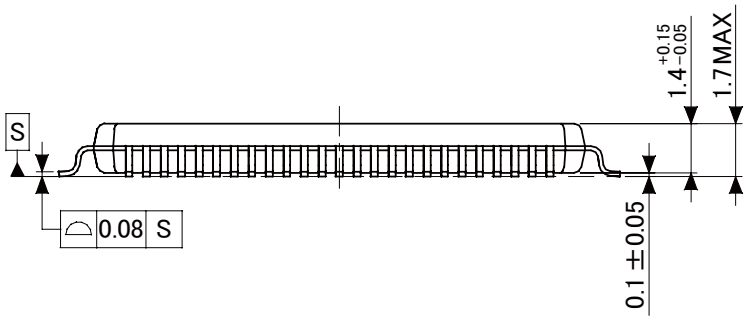


23.2 Type: P-LQFP100-1414-0.50H

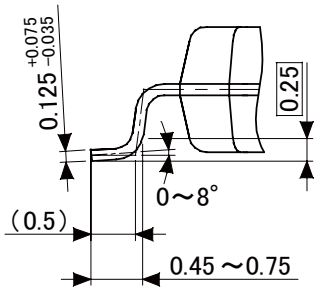
Dimensions

Unit: mm





Pin detail



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 NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, 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. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**