TOSHIBA

32 Bit RISC Microcontroller

TX03 Series

TMPM37AFSQG

TOSHIBA CORPORATION
Storage & Electronic Devices Solutions Company

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

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

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

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

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

Base Address = 0x0000_0000

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

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

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

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data description

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

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

d. Register description

Registers are described as shown below.

- Register name <Bit Symbol>

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

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

- Register name [Bit]

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

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

Revision History

Date	Revision	Comment
2016/03/18	1	First Release
2021/10/15	2	Contents Revised
2022/03/31	3	Contents Revised
2022/06/01	4	Contents Revised
2023/07/21	5	Contents Revised
2023/07/31	6	Contents Revised

Table of Contents

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

TMPM37AFSQG

2. Product Information

3. Processor Core

4. Memory Map

5. Reset Operation

6. Clock / Mode Control

7. Internal High-speed Oscillation Adjustment Function

8. Exceptions

9. Digital Noise Filter Circuit (DNF)

10. Input / Output Port

11. 16-bit Timer / Event Counters (TMRB)

12. Serial Channel with 4bytes FIFO (SIO/UART)

13. Serial Bus Interface (I2C/SIO)

14. 12-Bit Analog-to-Digital Converter

15. Motor Control Circuit (PMD: Programmable Motor Driver)

16. Vector Engine (VE+)

20. Voltage Detection Circuit (VLTD)

21. Oscillation Frequency Detector (OFD)

22. Watchdog Timer(WDT)

23. Flash Memory Operation

24. Debug Interface

25. Port Section Equivalent Circuit Schematic

26. Electrical Characteristics

27. Package Dimensions

CMOS 32-Bit Microcontroller

TMPM37AFSQG

TMPM37AFSQG is a Package product consisting of 32-bit RISC micro controller (MCU) and a three-phase pre-driver (MCD).

Outline and features of the TMPM37AFSQG are below:

1.1 Outline

1.1.1 MCU Unit

The MCU (Micro Controller Unit) is a 32-bit RISC microcontroller based on the ARM®Cortex®-M3 microprocessor.

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 = 32 bit) can be executed with one clock.
      - Division takes between 2 and 12 cycles depending on dividend and divisor

      [Low power consumption]
      - Optimized design using a low power consumption library
      - Standby function that stops the operation of the 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 : 4 KBytes
   - On-chip Flash ROM : 64 KBytes

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

4. Watchdog timer (WDT): 1 channel

Watchdog timer (WDT) generates a reset.

5. Power_On reset function (POR)

6. Voltage detect function (VLTD)

7. Oscillation frequency detect function (OFD)

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

9. Programmable motor driver (PMD): 1 channel
   - Three-phase complementary PWM generator
   - Synchronous AD convert start trigger generator
   - Emergency protective function

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

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

    I2C bus mode

12. 12-bit AD converter (ADC): 1 unit (5 channels)
    - Start by the internal trigger: TMRB interrupt/PMD trigger
    - Constant conversion mode
    - AD monitoring: 2 channels
    - Conversion speed 2.0 µsec (@ADC conversion clock = 40 MHz)

13. OP-Amp (AMP): 1 channel

    8 gains can be selected.

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

    I/O pins: 13 pins

15. Interrupt source
    - Internal 26 factors : The order of precedence can be set over 7 levels. (except the watchdog timer interrupt)
    - External 3 factors : The order of precedence can be set over 7 levels.

16. Standby mode

    Standby modes: IDLE, STOP

17. Clock generator (CG)
    - On-chip PLL (quadruple)
    - 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: 10 MHz

20. Maximum operating frequency : 40 MHz

1.1.2 MCD Unit

The MCD (Motor Control Driver) is a pre-driver that corresponds to the sine wave drive of three-phase brushless DC motors.

1. Three-phase full-wave drive

2. Motor power supply voltage

   Supports a 12V power supply and 24V power supply

3. Built-in regulator

4. Built-in over current detection circuit

1.1.3 Common Unit

1. Operating voltage range
   - VM=6 V to 32 V
   - DVDD5B = 4.5 V to 5.5 V (fsys=40 MHz)

     All functions operating
   - DVDD5 = 3.9 V to 4.5 V (fsys=40 MHz)

     (Except the 12-bit AD converter characteristics, electrical characteristics, and flash memory write characteristics)

2. Temperature range
   - -40 °C to 85 °C (except during Flash writing/erasing)
   - 0 °C to 70 °C (during Flash writing/erasing)

3. Package

   VQFN32 (5.0 mm x 5.0 mm, 0.5 mm pitch)

1.2 Block Diagram

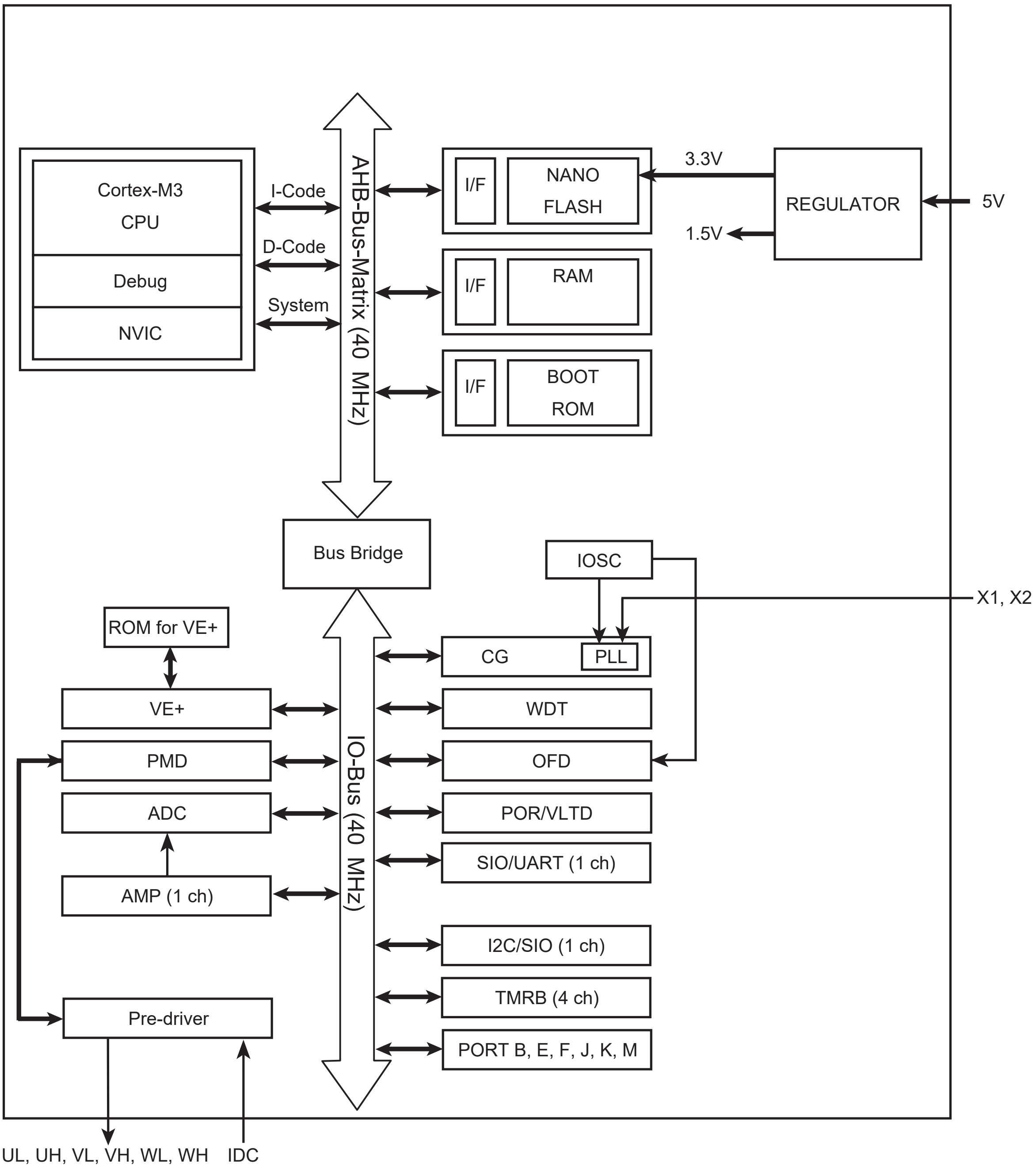


Figure 1-1 Block diagram of the TMPM37AFSQG (MCU)

1.3 Pin Layout (Top view)

The pin layout of TMPM37AFSQG is as shown below:

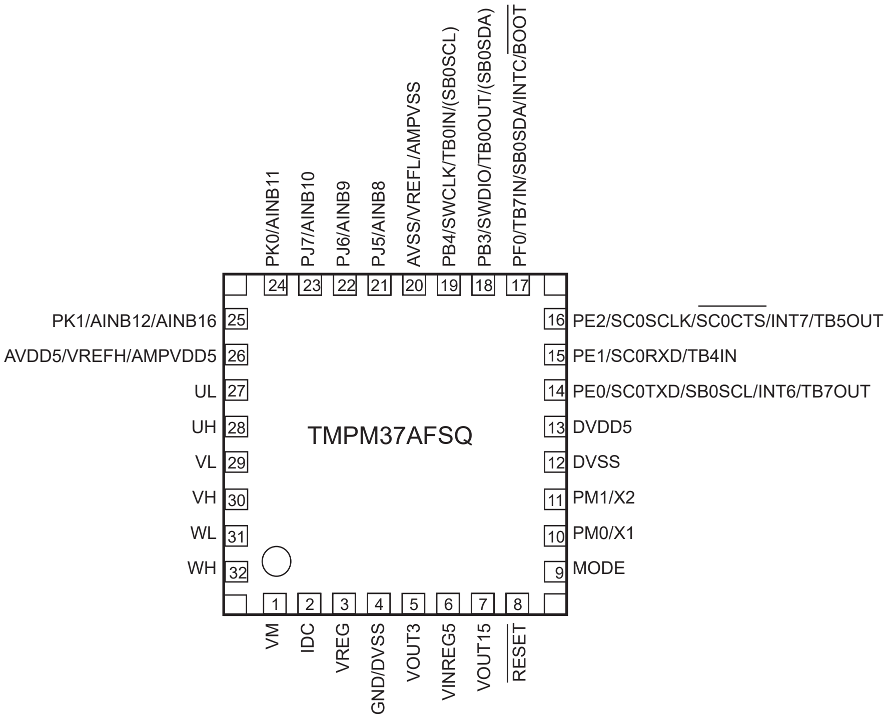


Figure 1-2 Pin Layout

1.4 Pin Names and Functions

1.4.1 Functional Pin Names and Functions

1.4.1.1 Peripheral Function Pin Names

Table 1-1 Peripheral functions and corresponding pin names and functions

Peripheral function	Pin name	Input or Output	Function
External interrupt	INTx	Input	External interrupt input pin. External interrupt pin has a noise filter (filter width: 30 ns typ.).
16-bit timer/event counter	TBxIN	Input	Input capture input pin.
	TBxOUT	Output	Output pin.
SIO/UART	SCxTXD	Output	Data output pin.
	SCxRXD	Input	Data input pin.
	SCxSCLK	I/O	Clock input/output pin.
	$\overline{\text{SCxCTS}}$	Input	Handshake input pin.
I2C/SIO	SBxSDA	I/O	Data input/output pin.
	SBxSCL	I/O	Clock input/output pin.
Analog-digital converter	AINBx	Input	Analog input pin.

1.4.1.2 Motor Control Pin Name

Table 1-2 Motor control pin names and functions

Debug pin name	Input or Output	Function
IDC	Input	MCD over current detection input pin.
UL	Output	U-phase low side output pin.
UH	Output	U-phase high side output pin.
VL	Output	V-phase low side output pin.
VH	Output	V-phase high side output pin.
WL	Output	W-phase low side output pin.
WH	Output	W-phase high side output pin.

1.4.1.3 Debug Pin Name

Table 1-3 Debug pin names and functions

Debug pin name	Input or Output	Function
SWDIO	I/O	Serial-wire data input/output pin.
SWCLK	Input	Serial-wire clock input pin.

1.4.1.4 Control Pin Name

Table 1-4 Control pin names and functions

Control pin name	Input or Output	Function
MODE	Input	Mode pin. Must be fixed to "Low" level.
$\overline{\text{RESET}}$	Input	Reset signal input pin (Note) With a pull-up circuit and noise filter (approximately 30 ns typ.)

1.4.1.5 Clock Pin Name

Table 1-5 Clock pin names and functions

Clock pin name	Input or Output	Function
X1	Input	High-speed oscillation connection pin.
X2	Output	High-speed oscillation connection pin.

1.4.1.6 Power Supply Pin Name

Table 1-6 Power supply pin names and functions

Power supply pin names	Function	Power supplied pin name
VM	Power supply pin for the MCD.	UH, UL, VH, VL, WH, WL
GND	GND pin for the MCD	
VREG	5V pin for the MCD	IDC, MODE
VOUT3	Capacitor (3.3 μF to 4.7 μF) connection pin for regulators.	
VOUT15	Capacitor (3.3 μF to 4.7 μF) connection pin for regulators.	
VINREG5	Power supply pin for regulators.	
DVDD5	Power supply pin for digital.	PB, PE, PF, PM, $\overline{\text{RESET}}$, MODE
DVSS	GND pin for digital	
AVDD5	Power supply pin for ADC unit. (Note 1)	PJ, PK
AVSS	GND pin for ADC. (Note 2)	
VREFHL	Analog reference voltage for ADC. (Note 1)	PJ, PK
VREFLH	Analog reference voltage for ADC. (Note 2)	
AMPVDD5	Power supply pin for Op-Amp. (Note 1	PJ, PK
AMPVSS	GND pin for Op-Amp. (Note 2)	

Note 1: This pin must be connected to power supply when the AD converter and Op-Amp are not used.

Note 2: This pin must be connected to GND when the AD converter and Op-Amp are not used.

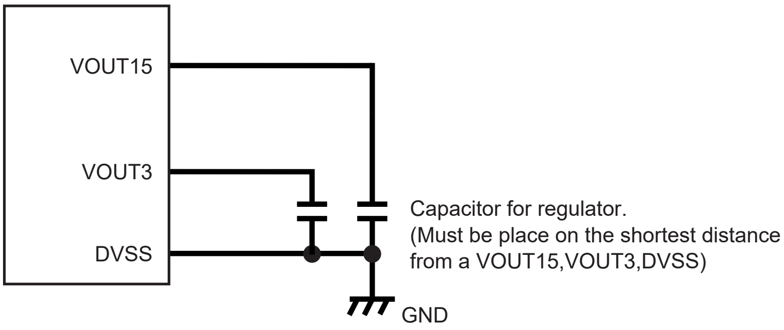

VOUT15
VOUT3
DVSS
Capacitor for regulator.
(Must be place on the shortest distance
from a VOUT15,VOUT3,DVSS)
GND

Figure 1-3 Capacitors connection diagram for regulators

1.4.2 Pin Names and the Functions of TMPM37AFSQG

1.4.2.1 Conventions Used in the Table

Various conventional symbols are used in the following tables.

1. Function A

   Dual functions in which the pins are assigned to ports without the function register settings are described.

2. Function B

   Dual functions in which the pins are assigned to ports by the register settings are described. The numbers shown in the "Function B" Column correspond to the numbers of function registers.

3. Pin specifications

   The symbols below have the following meanings:

   - SMT/CMOS: Input gate

     SMT: Schmitt trigger input

     CMOS: CMOS input

   - 5V_T: 5V tolerant

     -: Not applicable. This product group operates at 5 V.

   - OD: Programmable open-drain output

     Yes: Support

     N/A: Not available

   - PU/PD: Programmable pull-up/pull-down

     PU: Programmable pull-up is selectable.

     PD: Programmable pull-down is selectable.

1.4.2.2 Port/Debug Pins

Table 1-7 Pin numbers and functions <Sorted by PORT>

Pin No.	PORT	Function A	Function B						Port specification			
			1	2	3	4	5	6	PU/PD	OD	5V_T	SMT/CMOS
PORT B												
18	PB3		SWDIO	TB0OUT		(SB0SDA) (Note 1)			PU/PD	Yes	-	SMT
19	PB4		SWCLK	TB0IN		(SB0SCL) (Note 2)			PU/PD	Yes	-	SMT
PORT E												
14	PE0		SC0TXD	SB0SCL0 (Note 2)		INT6	TB7OUT		PU/PD	Yes	-	SMT
15	PE1		SC0RXD	TB4IN					PU/PD	Yes	-	SMT
16	PE2		SC0SCLK	$\overline{\text{SC0CTS}}$		INT7		TB5OUT	PU/PD	Yes	-	SMT
PORT F												
17	PF0	$\overline{\text{BOOT}}$	TB7IN	SB0SDA (Note 1)		INTC			PU/PD	Yes	-	SMT
PORT J												
21	PJ5	AINB8							PU/PD	Yes		SMT
22	PJ6	AINB9							PU/PD	Yes	-	SMT
23	PJ7	AINB10							PU/PD	Yes	-	SMT
PORT K												
24	PK0	AINB11							PU/PD	Yes	-	SMT
25	PK1	AINB12/ AINB16 (Note 3)							PU/PD	Yes	-	SMT
PORT M												
10	PM0	X1							PU/PD	Yes	-	SMT
11	PM1	X2							PU/PD	Yes	-	SMT

Note 1: SB0SDA is toggled between PB3 and PF0. (PF0 gains a higher priority.)

Note 2: SB0SCL is toggled between PB4 and PE0. (PE0 gains a higher priority.)

Note 3: AINB16 is connected to the external device via the internal Op-AMP.

1.4.3 Control Pins

Table 1-8 Pin numbers and Pin names

Pin No.	Control Pin name
8	$\overline{\text{RESET}}$
9	MODE

1.4.4 Motor Control Pins

Table 1-9 Pin numbers and Pin names

Pin No.	Control Pin name
2	IDC
27	UL
28	UH
29	VL
30	VH
31	WL
32	WH

1.4.5 Power Supply Pins

Table 1-10 Power supply pins

Pin No.	Type	Voltage range
1	VM	6 to 32 V
13	DVDD5	4.5 to 5.5 V
26	AVDD5/VREFH/AMPVDD5	
6	VINREG5	

Table 1-11 Output dedicated pins

Pin No.	Pin names	Voltage range
3	VREG	VREG must be connected to GND through 4.7 μF capacitor.
7	VOUT15	VOUT15 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit.
5	VOUT3	VOUT3 must be connected to DVSS through 3.3 to 4.7μF capacitor for supply power to internal circuit.

Note: VOUT15 and VOUT3 are the capacitor connection pins that are used to stabilize the internal regulator outputs. These pins do not supply power to the external circuits.

2. Product Information

This chapter describes peripheral function-related channels or number of units, information of pins and product specific function information. Use this chapter in conjunction with Chapter Peripheral Function.

- "2.1.1 16-bit Timer / Event Counter (TMRB)"
- "2.1.2 Serial Channel (SIO/UART)"
- "2.1.3 Serial Bus Interface (I2C/SIO)"
- "2.1.4 Vector Engine (VE+)"
- "2.1.5 Motor Control Circuit (PMD: Programmable Motor Driver)"
- "2.1.6 Analog/Digital Converter (ADC)"
- "2.1.7 Watchdog Timer (WDT)"
- "2.1.8 Debug Interface"

2.1 Information of Each Peripheral Function

2.1.1 16-bit Timer / Event Counter (TMRB)

TMPM37AFSQG incorporates 4 channels of the TMRB.

Table 2-1 Pin specifications

Channel	TBxOUT	TBxIN
TMRB0	PB3	PB4
TMRB4	-	PE1
TMRB5	PE2	-
TMRB7	PE0	PF0

Note: Use any one of combinations.

Table 2-2 Interrupts

Channel	TBxOUT	Input capture	TMRB interrupt
TMRB0	PB3	INTCAP00, INTCAP01	INTTB00, INTTB01
TMRB4	-	INTCAP40, INTCAP41	INTTB40, INTTB41
TMRB5	PE2	-	INTTB50, INTTB51
TMRB7	PE0	INTCAP70, INTCAP71	INTTB70, INTTB71

2.1.2 Serial Channel (SIO/UART)

TMPM37AFSQG incorporates 1 channel of the SIO.

Table 2-3 Pin specifications

Channel	SCxTXD	SCxRXD	SCxSCLK	$\overline{\text{SCxCTS}}$
SC0	PE0	PE1	PE2	PE2

Table 2-4 Interrupts

Channel	Serial reception	Serial transmission
SC0	INTRX0	INTTX0

Table 2-5 Internal connection specification

Channel	Transfer clock input
SC0	TMRB4

2.1.3 Serial Bus Interface (I2C/SIO)

TMPM37AFSQG incorporates 1 channel of the I2C, and the SIO is not supported.

Table 2-6 Pin specification

Channel	SBxSDA	SBxSCL
I2C0	PF0 (PB3)	PE0 (PB4)

Table 2-7 Interrupt

Channel	Interrupt
I2C0	INTSBI0

2.1.4 Vector Engine (VE+)

TMPM37AFSQG incorporates 1 channel of built-in VE.

Table 2-8 Interrupt

Channel	Interrupt
VE	INTVCNB

Table 2-9 Internal connection specifications

Channel	PWM interrupt signal input	AD conversion completion signal input	AD conversion result Inputs
VE	INTPMD1	INTADBPDB	ADBREG0 to 3

2.1.5 Motor Control Circuit (PMD: Programmable Motor Driver)

TMPM37AFSQG incorporates 1 channel of the PMD.

Table 2-10 Pin specifications

Channel	$\overline{\text{OVVx}}$ Over voltage status detection input	$\overline{\text{EMGx}}$ Emergency status detection input	ZOx Z-phase output pin	WOx W-phase output pin	YOx Y-phase output pin	VOx V-phase output pin	XOx X-phase output pin	UOx U-phase output pin
PMD1	IDC	IDC	WL	WH	VL	VH	UL	UH

Table 2-11 Interrupts

Channel	OVV interrupt	EMG interrupt	PWM interrupt
PMD1	INTOVV1	INTEMG1	INTPMD1

Table 2-12 Internal connection specifications (1/2)

Channel	EMG protection release input (VE)	PWM compare input (VE)	Output control input (VE)	Trigger compare input (VE)	Select trigger input (VE)	OVV input (ADC)
PMD1	VEEMGRS1	VECMPU1, V1, W1	VEOUTCR1	VETRGCMP10, 11	VETRGSEL1	ADCB monitoring function compare 0, 1 output

Table 2-13 Internal connection specifications (2/2)

Channel	MDOUT transfer timing signal input
PMD1	INTTB00

2.1.6 Analog/Digital Converter (ADC)

TMPM37AFSQG incorporates 1 unit of ADC (a 12-bit successive-approximation analog-to-digital converter). The ADC operates in conjunction with the vector engine and the PMD circuit to support vector control for motors.

Table 2-14 Pin specifications

Unit	AINB8 to 10	AINB11 to 12	AINB16
ADCB	PJ5 to 7	PK0 to 1	PK1

Table 2-15 Interrupts

Unit	AD conversion triggered by completion of the PMD	AD conversion triggered by completion of the timer	AD conversion started by completion of software	AD conversion monitoring function
ADCB	INTADBPDB	INTADBTMR	INTADBSFT	INTADBCPA INTADBCPB

Table 2-16 Internal connection specifications

Unit	PMD1 trigger input	TMRTRG
ADCB	PMD1TRG0 to 5	INTTB51

2.1.7 Watchdog Timer (WDT)

In TMPM37AFSQG, bit 2 <I2WDT> of the WDMOD register (Watchdog Timer Mode Register) is not supported; therefore make sure to write "0".

2.1.8 Debug Interface

TMPM37AFSQG supports serial wire debug ports.

Table 2-17 Pin specifications

	SWDIO	SWCLK
Serial wire	PB3	PB4

3. 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.

3.1 Information on the processor core

The following table shows the revision of the processor core in the TMPM37AFSQG.

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
TMPM37AFSQG	r2p1

3.2 Configurable Options

The Cortex-M3 core has optional blocks. The optional blocks of the revision r2p1 are ETM™ and MPU. The following table shows the configurable options in the TMPM37AFSQG.

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

3.3 Exceptions/ Interruptions

Exceptions and interruptions are described in the following section.

3.3.1 Number of Interrupt Inputs

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

TMPM37AFSQG has 32 interrupt inputs. The number of interrupt inputs is reflected in <INTLINESNUM [4:0]> bit of NVIC register. In this product, if read <INTLINESNUM[4:0]> bit, 0x00 is read out.

3.3.2 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.

TMPM37AFSQG 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.

3.3.3 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.

3.3.4 SYSRESETREQ

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

TMPM37AFSQG provides the same operation when SYSRESETREQ signal are output.

3.3.5 LOCKUP

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

TMPM37AFSQG does not use this signal. To return from LOCKUP status, it is necessary to use non-maskable interrupt (NMI) or reset.

3.3.6 Auxiliary Fault Status register

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

However, TMPM37AFSQG is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

3.4 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.

TMPM37AFSQG does not use event output signals and event input signals. Please do not use SEV instruction and WFE instruction.

3.5 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.

TMPM37AFSQG 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."

3.6 Exclusive access

In Cortex-M3 core, the DCode bus system supports exclusive access. However TMPM37AFSQG does not use this function.

4. Memory Map

4.1 Memory Map

The memory maps for TMPM37AFSQG are based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM37AFSQG 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.

4.1.1 TMPM37AFSQG Memory Map

Figure 4-1 shows the memory map of the TMPM37AFSQG.

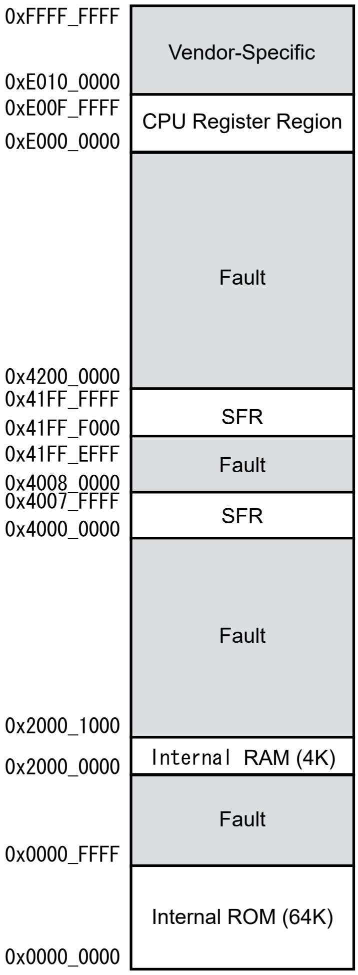


Figure 4-1 Memory Map

4.2 Details of SFR area

Table 4-1shows the details of the SFR area.

Do not access a reserved area in Table 4-1. See the chapter of each peripheral function for details.

Table 4-1 Details of SFR

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

Table 4-2 Address lists of peripheral functions

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

Table 4-2 Address lists of peripheral functions

Port G	0x4000_0180	PORT
Port J	0x4000_0240	PORT
Port K	0x4000_0280	PORT
Port M	0x4000_0300	PORT

TMRB 0	0x4001_0000	TMRB
TMRB 4	0x4001_0100	TMRB
TMRB 5	0x4001_0140	TMRB
TMRB 7	0x4001_01C0	TMRB

SBI 0	0x4002_0000	SBI

SIO UART 0	0x4002_0080	SIO / UART

ADC	0x4003_0200	ADC

AMP D	0x4003_0418	AMP

WDT	0x4004_0000	WDT

CG	0x4004_0200	CG

OSCTRIM	0x4004_0300	TRM

OFD	0x4004_0800	OFD

LVD	0x4004_0900	LVD

VE	0x4005_0000	VE

PMD 1	0x4005_0480	PMD

DNF	0x4006_0000	DNF

5. Reset Operation

The following are sources of reset operation.

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

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

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

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

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

5.1 Cold Reset

When turning-on power, it is necessary to take a stable time of built-in regulator, built-in Flash memory and internal high-speed oscillator into consideration. TMPM37AFSQG has a function to insert a stable time automatically.

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

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

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

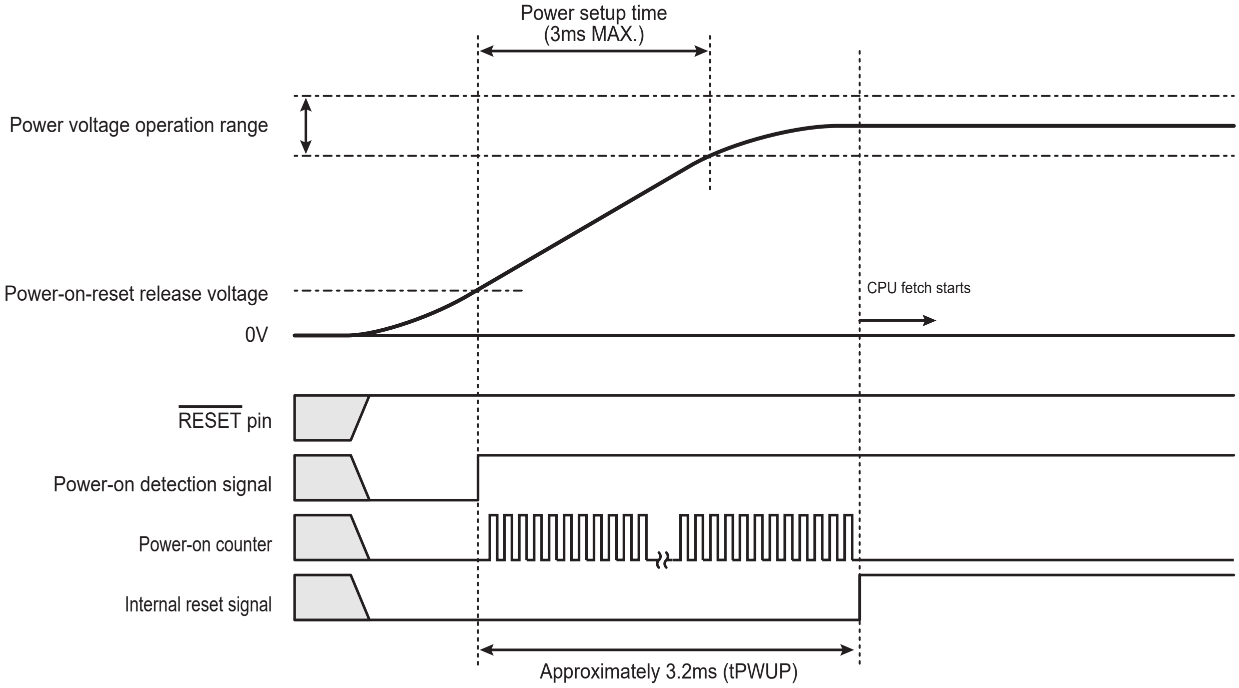

Power setup time
(3ms MAX.)
Power voltage operation range
Power-on-reset release voltage
0V
CPU fetch starts
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal
Approximately 3.2ms (tPWUP)

Figure 5-1 Reset Operation by Power-on Circuit

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

The reset using the $\overline{\text{RESET}}$ pin will be effective after the power-on counter finishes. And if $\overline{\text{RESET}}$ pin is set to "High" within 3.2ms after power-on reset signal becomes "High", the reset process will be the same as the power-on described in 5.1.1.

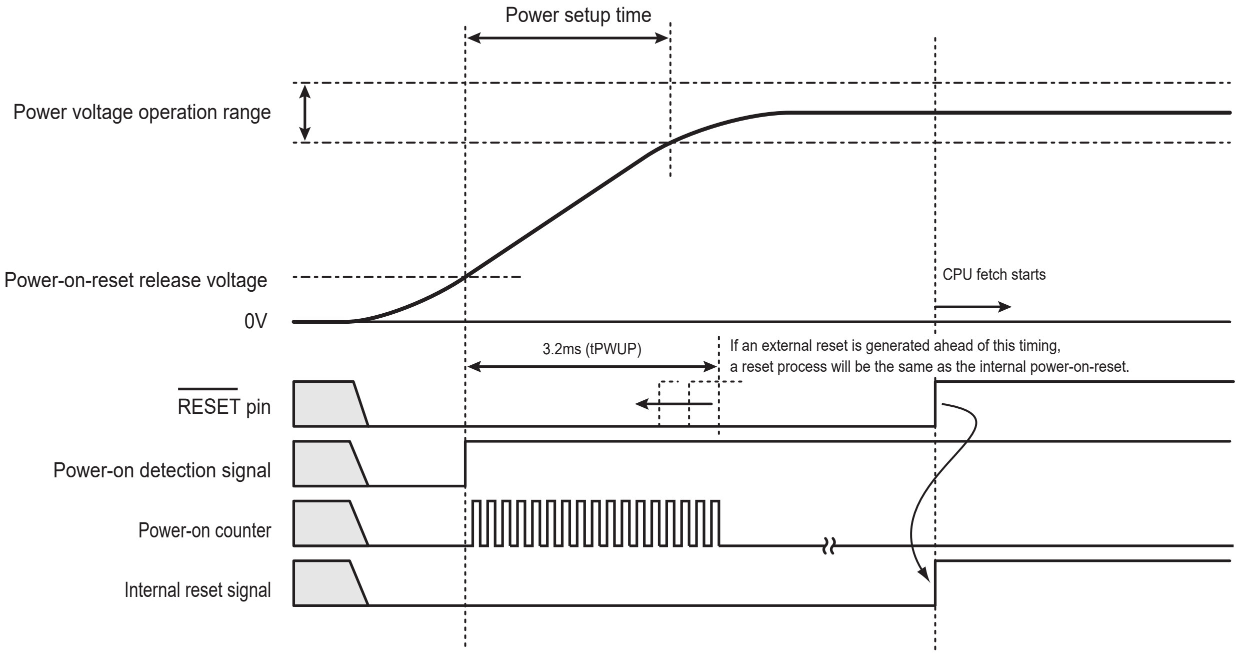

Power setup time
Power voltage operation range
Power-on-reset release voltage
0V
CPU fetch starts
3.2ms (tPWUP)
If an external reset is generated ahead of this timing,
a reset process will be the same as the internal power-on-reset.
RESET pin
Power-on detection signal
Power-on counter
Internal reset signal

Figure 5-2 Reset Operation by $\overline{\text{RESET}}$ pin

5.2 Warm-up

5.2.1 Reset Duration

To do reset TMPM37AFSQG, the following condition is required; power supply voltage is in the operational range; RESET pin is kept "Low" at least for 12 system clocks by internal high frequency oscillator. After $\overline{\text{RESET}}$ pin becomes "High", internal reset will be released.

5.3 After reset

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

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

6. Clock / Mode Control

6.1 Features

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

There is also 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 TMPM37AFSQG can operate in six types of low power mode to reduce power consumption according to its usage conditions.

6.2 Registers

6.2.1 Register List

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

Base Address = 0x4004_0200

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

6.2.2 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

6.2.3 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.

Bit	Bit Symbol	Type	Function
1	WUEF	R	Status of warm-up timer (WUP) (Note1)(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 (Note1)(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 cannot 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 ini-tial value, the <HOSCON> cannot 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.

6.2.4 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 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 STOP mode, disable the oscillation (OSC1 or OSC2) which is unused as system clock.

6.2.5 CGPLLSEL (PLL Selection Register)

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

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

6.3 Clock control

6.3.1 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 multiplied by PLL (x 5 or 4)
fc	: Clock specified by CGPLLSEL<PLLSEL> (high-speed clock)
fgear	: Clock specified by CGSYSCR<GEAR[2:0]>
fsys	: The same clock as fgear (system clock)
fperiph	: Clock specified by CGSYSCR<FPSEL>
φT0	: Clock specified by CGSYSCR<PRCK[2:0]> (Prescaler clock)

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

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

6.3.2 Initial Values after Reset

Reset operation initializes the clock configuration as follows.

External high-speed oscillator (OSC1)	: Stop (X1,X2)
Internal high-speed oscillator (OSC2)	: 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}.

$f_C = {}_{OSC2}$

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

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

$φT0 = f_{periph}$ ($= f_{OSC2}$)

6.3.3 Clock system Diagram

Figure 6-1 shows the clock system diagram.

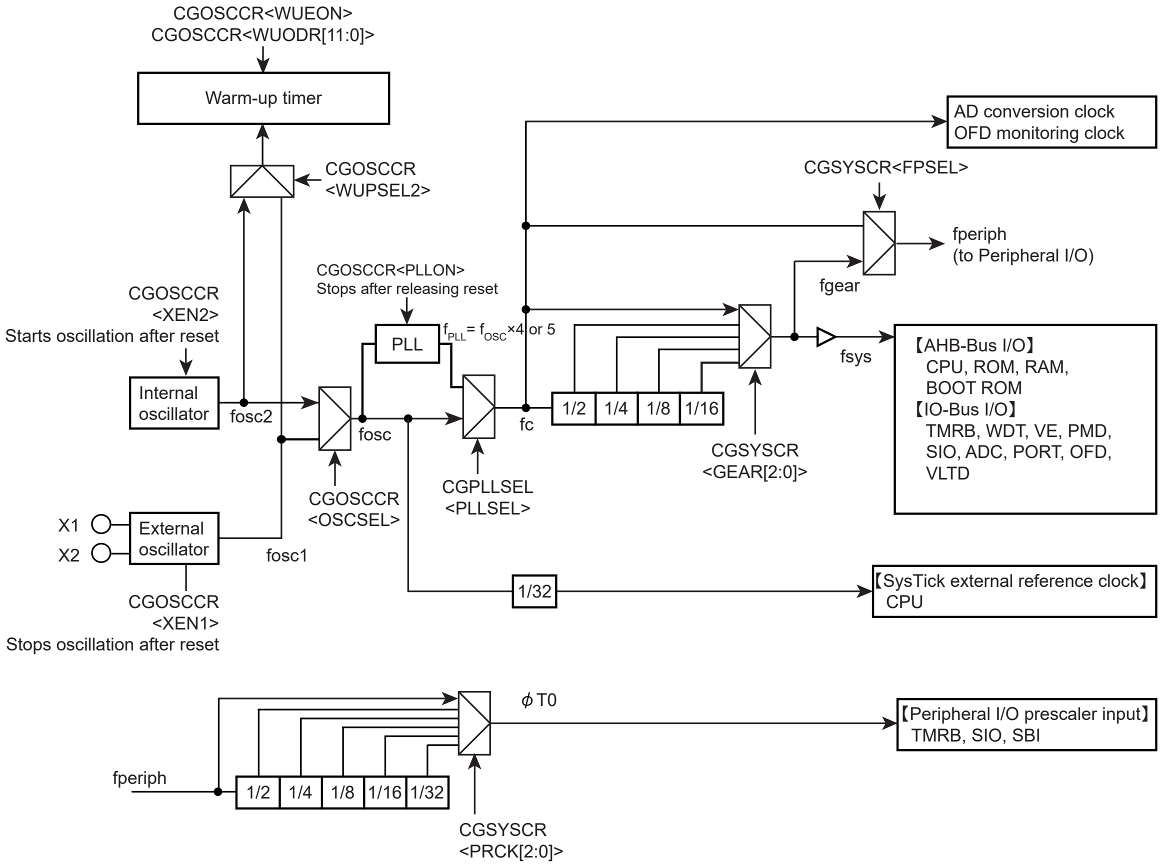


Figure 6-1 Clock Block Diagram

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

6.3.4 Clock Multiplication Circuit (PLL)

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

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

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

6.3.4.1 Stability time

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

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

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

6.3.4.2 The sequence of PLL setting

The following shows PLL setting sequence after reset.

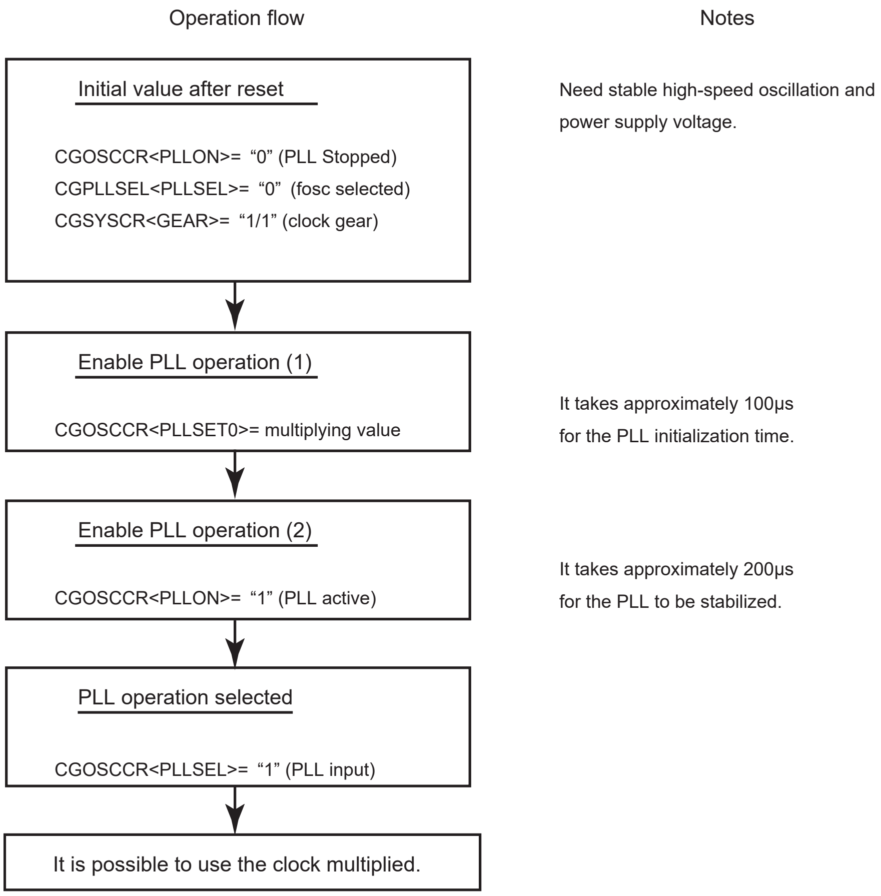

Operation flow
Notes
Initial value after reset
CGOSCCR<PLLON>= “0” (PLL Stopped)
CGPLLSEL<PLLSEL>= “0” (fosc selected)
CGSYSCR<GEAR>= “1/1” (clock gear)
Need stable high-speed oscillation and power supply voltage.
Enable PLL operation (1)
CGOSCCR<PLLSET0>= multiplying value
It takes approximately 100µs for the PLL initialization time.
Enable PLL operation (2)
CGOSCCR<PLLON>= “1” (PLL active)
It takes approximately 200µs for the PLL to be stabilized.
PLL operation selected
CGOSCCR<PLLSEL>= “1” (PLL input)
It is possible to use the clock multiplied.

Figure 6-2 PLL setting sequence after reset

Note: When the PLL is attempted to be stopped, check whether CGPLLSEL<PLLSEL> is "0". After that, check whether CGPLLSEL<PLLSEL> is "0". Then, set "0" to CGOSCCR<PLLON> (PLL stop).

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

6.3.5 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 "6.6.6 Warm-up".

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

In this case, an interrupt for returning from 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.

How to configure the warm-up function.

1. Specify the count up clock

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

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

2. Specify the warm-up counter value

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

   The following shows the warm-up setting and example.

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

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

$$\boxed{\frac{\text{Setting value of warm-up time}}{\text{Input cycle by frequency(s)}} = \frac{\text{5ms}}{\text{1/8MHz}} = \text{40,000cycles} = \text{0x9C40}}$$

   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 6.3.6 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)

6.3.6 System Clock

The TMPM37AFSQG 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 6-1 Range of high-speed frequency (unit : MHz)

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

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

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

Note 3: Do not use 1/16 when "PLL =OFF" is used.

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

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

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

Note that when CGOSCCR<HOSCON> is set to "1", all registers of the port M are prohibited to be accessed. Therefore, the value of the port M cannot be modified.

6.3.7 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: When the clock gear is used, specify the time for the prescaler output Tn, which is from each peripheral function, to be slower than fsys (Tn < fsys). Do not switch the clock gear while the timer counter or other peripheral functions are operating.

6.4 Modes and Mode Transitions

6.4.1 Mode Transitions

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

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

Figure 6-3 shows mode transition diagram.

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

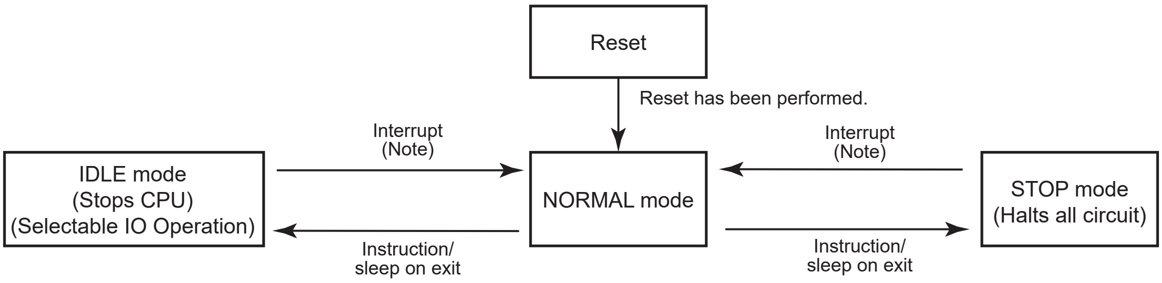


Figure 6-3 Mode Transition Diagram

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 "6.6.6 Warm-up".

6.5 Operation Mode

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

6.5.1 NORMAL mode

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

It is shifted to NORMAL mode after reset.

6.6 Low Power Consumption Modes

The TMPM37AFSQG has two low power consumption modes: IDLE and STOP. To shift to 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 TMPM37AFSQG does not offer any event for releasing low power consumption mode. Transition to low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.

Note 2: The TMPM37AFSQG does not support low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

The features of each mode are described as follows.

6.6.1 IDLE Mode

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

The following peripheral functions can be enabled or disabled in 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.

6.6.2 STOP mode

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

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

STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>. Table 6-2 shows the pin status in STOP mode.

Table 6-2 Pin States in STOP mode

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

o : Input or output enabled.

× : Input or output disabled.

6.6.3 Low power Consumption Mode Setting

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

Table 6-3 shows the mode setting in the <STBY[2:0]>.

Table 6-3 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]>.

6.6.4 Operational Status in Each Mode

Table 6-4 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 6-4 Operational Status in Each Mode

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

o : Operating

× : Stopped

Note 1: It depends on CGSTBYCR<DRVE>.

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

6.6.5 Releasing the Low Power Consumption Mode

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 low power consumption mode selected.

Details are shown in Table 6-5.

Table 6-5 Release Source in Each Mode

Low power consumption mode			IDLE (programmable)	STOP
Release source	Interrupt	INT6, 7, C (Note1)	o	o
		INTRX0, 1, INTTX0, 1	o	×
		INTVCNB	o	×
		INTEMG1	o	×
		INTOVV1	o	×
		INTADBPDB	o	×
		INTTB00, 40, 50, 70 INTTB01, 41, 51, 71	o	×
		INTPMD1	o	×
		INTCAP00, 50, 70 INTCAP01, 51, 71	o	×
		INTADBCPA, INTADBCPB	o	×
		INTADBSFT	o	×
		INTADBTMR	o	×
		INTSBI0	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 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 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 "6.6.6 Warm-up" about warm-up time.

- Release by interrupt request

  To release 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 STOP modes. And the digital noise filter circuit should be set to disable as well.

- Release by Non-Maskable Interrupt (NMI)

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

Note: Note that the WDT cannot be cleared by the CPU operation in IDLE mode.

- Release by reset

  All low power consumption modes can be released by reset from the $\overline{\text{RESET}}$ pin. After that, the mode switches to 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.

For details of the interrupt, refer to "Interrupts" in "Exceptions" chapter.

6.6.6 Warm-up

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

In the mode transition from STOP to 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 STOP mode.

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

Note 2: 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. 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 6-6 shows whether the warm-up setting of each mode transition is required or not.

Table 6-6 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

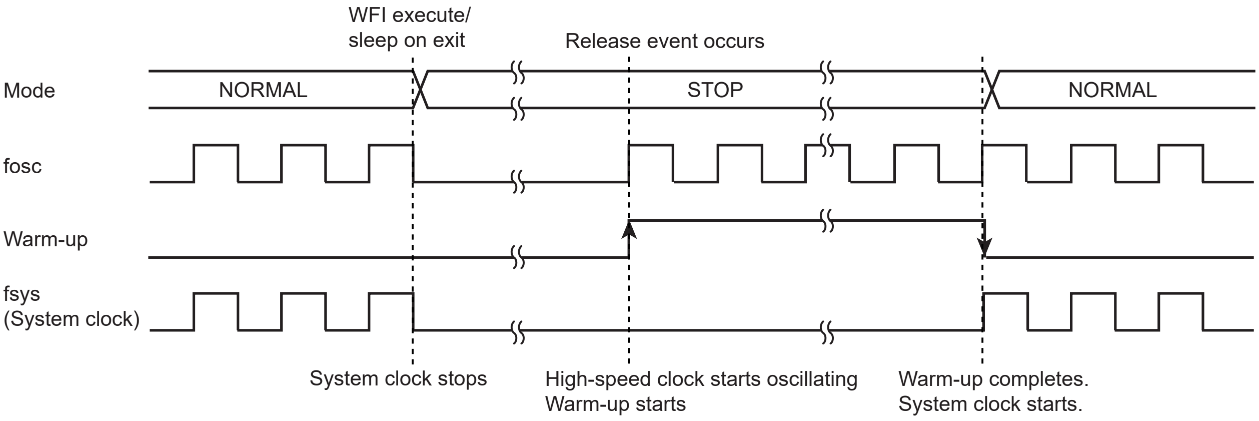
6.6.7 Clock Operation in Mode Transition

The clock operation in mode transition are described Chapter 6.6.7.1.

6.6.7.1 Transition of operation modes : NORMAL → STOP → NORMAL

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

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



7. Internal High-speed Oscillation Adjustment Function

TMPM37AFSQG has the internal high-speed oscillation adjustment function.

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

7.1 Structure

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

Figure 7-1 shows the function configuration.

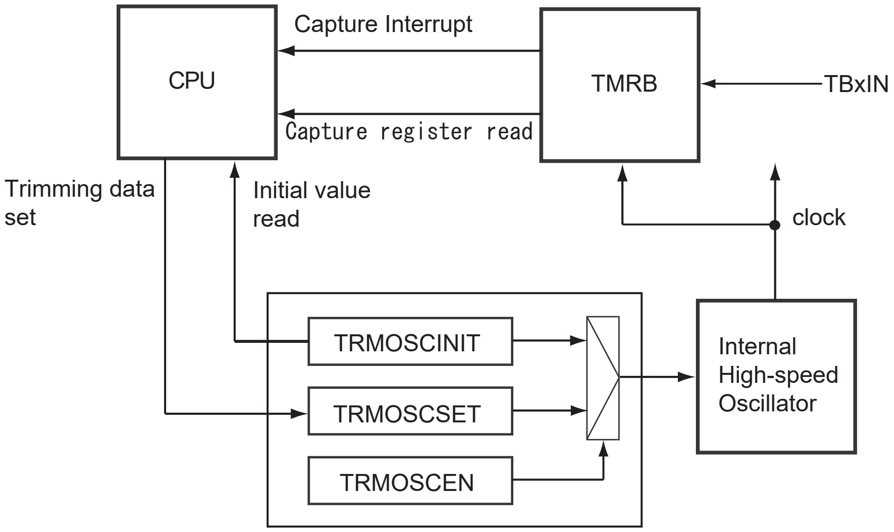


Figure 7-1 Function block diagram

7.2 Registers

7.2.1 Register list

The control registers and its addresses are as follows.

Base Address = 0x4004 _ 0300

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

7.2.2 TRMOSCPRO (Protect register)

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

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

7.2.3 TRMOSCEN (Enable register)

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

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

7.2.4 TRMOSCINIT (Initial trimming value monitor register)

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

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

For details about the specific setting and adjustment value of coarse trimming and fine trimming, refer to "Table 7-1 Adjustment range".

7.2.5 TRMOSCSET (Trimming value setting register)

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

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

For details about the specific setting and adjustment value of coarse trimming and fine trimming, refer to "Table 7-1 Adjustment range".

7.3 Operational Description

7.3.1 Outline

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

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

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

7.3.2 Adjustment range

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

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

Table 7-1 Adjustment range

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

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

7.3.3 Internal Oscillation Frequency Measurement using TMRB

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

8. Exceptions

This chapter describes the features, types, and handling of exceptions.

Exceptions have close relationship to the CPU core. Refer to "ARM documentation set for the ARM Cortex-M3" if needed.

8.1 Overview

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

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

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

8.1.1 Exception Types

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

For detailed descriptions on each exception, refer to "ARM documentation set for the ARM Cortex-M3".

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

8.1.2 Handling Flowchart

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

Each step is described later in this chapter.

Processing	Description	See
Detection by CG/CPU	The CG/CPU detects the exception request.	Section 8.1.2.1
⇩		
Handling by CPU	The CPU handles the exception request.	Section 8.1.2.2
⇩		
Branch to ISR	The CPU branches to the corresponding interrupt service routine (ISR).	
⇩		
Execution of ISR	Necessary processing is executed.	Section 8.1.2.3
⇩		
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 8.1.2.4

Note: ISR: Interrupt Service Routine

8.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

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

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

An interrupt request is generated from an external interrupt pin or peripheral function. For interrupts that are used for releasing a standby mode, relevant settings must be made in the clock generator. For details, refer to "8.5 Interrupts".

(2) Exception detection

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

Table 8-1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 8-1 Exception Types and Priority

No.	Exception type	Priority	Description
1	Reset	−3 (highest)	Reset pin, WDT, POR, VLTD, OFD or 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 to 10	Reserved	−	
11	SVCall	Configurable	System service call with SVC instruction
12	Debug Monitor	Configurable	Debug monitor when the CPU is not faulting
13	Reserved	−	
14	PendSV	Configurable	Pendable system service request
15	SysTick	Configurable	Notification from the 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 "8.5.2 List of Interrupt Sources".**

(3) Priority setting

- Priority levels

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

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

In TMPM37AFSQG, <PRI_n> is a 3-bit configuration.

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 numbers have the higher priorities.

- Priority grouping

The priority levels 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 their pre-emption priority levels are the same, then they are compared with the sub priority. If their sub priority levels are the same, the smaller the exception numbers have the higher the priorities.

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

Table 8-2 Priority grouping setting

<PRIGROUP[2:0]> setting	<PRI_n[7:0]>		Number of pre-emption priorities	Number of sub-priorities
	Pre-emption field	Sub-priority field		
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
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 in <PRI_n[7:5]> and <PRI_n[4:0]> is "00000".

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

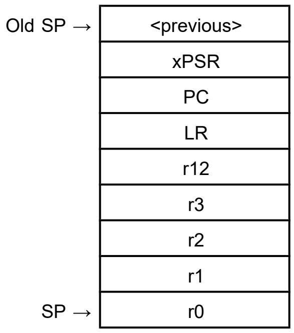
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine (ISR). This is called "pre-emption".

(1) Stacking

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

1. Program Counter (PC)
2. Program Status Register (xPSR)
3. r0 - r3
4. r12
5. Link Register (LR)

The SP is decremented by eight words on 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 performs an instruction fetch in the interrupt service routine simultaneously with storing data to the register.

Prepare a vector table containing the start addresses of ISRs of 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.

Make sure to always set the first four words (initial value of the stack address, reset ISR address, NMI ISR address, and Hard Fault ISR address).

For other exceptions, set ISR addresses if necessary.

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

8.1.2.3 Executing an ISR

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

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

For details about interrupt handling, see "8.5 Interrupts".

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

8.1.2.4 Exception Exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

In this case, the CPU skips state saving and restoration between interrupts. 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) Return sequence

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

- Restoring the registers.

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

- Loading the current active interrupt number

  Loads the current active interrupt number from the preserved xPSR. The CPU uses this to determine the interrupt at which the CPU should return.

- Selecting the SP

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

8.2 Reset Exceptions

Reset exceptions are generated from the following six sources.

Use the Reset Flag (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 "Power on Reset Circuit (POR)" for detail.

- Reset exception by VLTD

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

- Reset exception by OFD

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

- Reset exception by WDT

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

- Reset exception by SYSRESETREQ

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

8.3 Non-Maskable Interrupts (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.

8.4 SysTick

SysTick is a interrupt function using the CPU's system timer.

When the value is set in the SysTick Reload Value Register and it is enabled with 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 can check whether an exception is pended, and whether the flag is set when the timer reaches "0".

Note: In this product, fosc (selected by CGOSCCR <OSCSEL> <HOSCON>) is divided by 32 and is used as external reference clock.

8.5 Interrupts

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

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

It sets the 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 releasing low power consumption mode logic in CG. Therefore, appropriate settings according to the source are required in the releasing low power consumption mode logic.

8.5.1 Interrupt Sources

8.5.1.1 Interrupt Route

Figure 8-1 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 low power consumption mode (route 2) are input to the logic for releasing low power consumption mode in CG. After the active level for each interrupt request is detected, the logic for releasing low power consumption mode makes the active level to the new interrupt request signal. This signal is input to CPU. (route 6, 7 and 8).

The interrupt request of an external interrupt pins (route 3) is used as the either releasing or no releasing source of low power consumption mode by <INTxEN>.

The interrupt request of an external interrupt pin used as releasing low power consumption mode is input to the logic for releasing low power consumption mode. If the specified active level is detected, the logic for releasing low power consumption mode convert the active level signal to the interrupt request, the interrupt request is input to CPU (route 2, 4, 5).

The interrupt request of an external interrupt pin not used as releasing the low power consumption mode is input CPU directly (route 2, 3, 5).

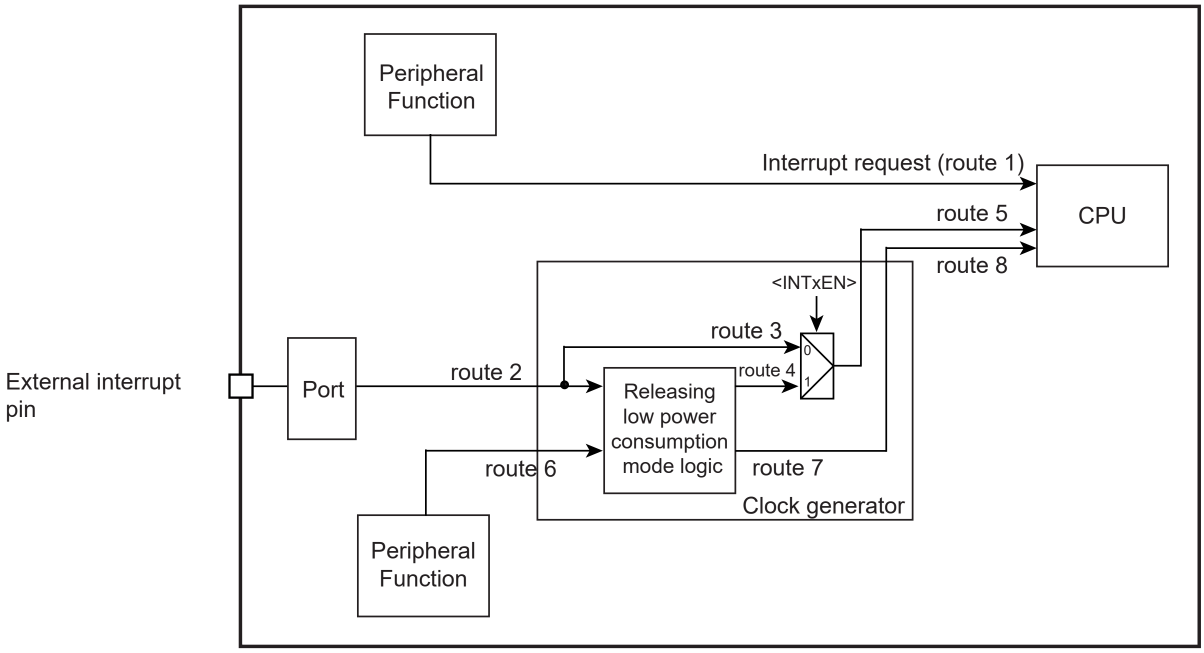


Figure 8-1 Interrupt route

8.5.1.2 Generation of the Interrupt Request

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

- From external interrupt pins

  When the external interrupt pin is used, set the port control register so that the external interrupt pin can perform as an interrupt function pin.
- From peripheral functions

  When the interrupt function of the peripheral functions is used, 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 generation)

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

The CPU recognizes the "High" level of the interrupt request signal as interrupt request.

8.5.1.3 Setting the Registers for Releasing Low Power Consumption Mode

Some interrupt request can be used to release the low power consumption mode.

To use the interrupt request for releasing low power consumption mode, set "1" to <INTxEN>, and specify the active level for releasing low power consumption mode using <EMCGx[2:0]>.

When the level specified by <EMCGx[2:0]> for the releasing low power consumption mode is input into an external interrupt pin, low power consumption mode is released, and the interrupt of "High" level is generated.

When <EMCGx[2:0]> is "100", the active level detected until low power consumption mode is released can be read from <EMSTx[1:0]>.

Clearing the interrupt request is specified with CGICRCG<ICRCG>. <EMSTx[1:0]> is initialized to "00" by clearing the interrupt request using CGICRCG<ICRCG>.

8.5.2 List of Interrupt Sources

Table 8-3 shows the list of interrupt sources.

Table 8-3 List of Interrupt Sources

No.	Interrupt Source		The active level to release low power consumption mode					CG interrupt mode control register
			"Low" level	"High" level	Rising edge	Falling edge	Both edge	
0	INTRX0	Serial reception (channel 0)						
1	INTTX0	Serial transmit (channel 0)						
2	Reserved	-						
3	Reserved	-						
4	INTVCNB	Vector Engine interrupt B						
5	INTEMG1	PMD1 EMG interrupt						
6	INTOVV1	PMD1 OVV interrupt						
7	INTADBPDB	ADC unit B conversion triggered by completion of PMD1						
8	INTTB00	16-bit TMRB0 compare match detection 0/ Overflow						
9	INTTB01	16-bit TMRB0 compare match detection 1						
10	INTTB40	16-bit TMRB4 compare match detection 0/ Overflow						
11	INTTB41	16-bit TMRB4 compare match detection 1						
12	INTTB50	16-bit TMRB5 compare match detection 0/ Overflow						
13	INTTB51	16-bit TMRB5 compare match detection 1						
14	INTPMD1	PMD1 PWM interrupt						
15	INTCAP00	16-bit TMRB0 input capture 0						
16	INTCAP01	16-bit TMRB0 input capture 1						
17	INTCAP40	16-bit TMRB4 input capture 0						
18	INTCAP41	16-bit TMRB4 input capture 1						
19	INT6	External interrupt pin 6	o	o	o	o	o	CGIMCGA
20	INT7	External interrupt pin 7	o	o	o	o	o	
21	INTADBCPA	ADC unit B conversion monitoring function interrupt A						
22	INTADBCPB	ADC unit B conversion monitoring function interrupt B						
23	INTADBSFT	ADC unit B conversion started by completion of software						
24	INTADBTMR	ADC unit B conversion triggered by completion of timer						
25	Reserved	-						
26	INTTB70	16-bit TMRB7 compare match detection 0/ Overflow						
27	INTTB71	16-bit TMRB7 compare match detection 1						
28	INTCAP70	16-bit TMRB7 input capture 0						
29	INTCAP71	16-bit TMRB7 input capture 1						
30	INTC	External interrupt pin C	o	o	o	o	o	CGIMCGA
31	INTSBI0	Serial bus interface						

Note: The active level marked with "o" can be used to release the low power consumption mode.

8.5.3 Interrupt Handling

8.5.3.1 Flowchart

The following shows how an interrupt is handled.

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

Processing	Details	See
Settings for detection Settings for sending interrupt signal	Set the relevant NVIC registers for detecting interrupts. o Common setting NVIC registers Execute an appropriate setting to send the interrupt signal depending on the interrupt type. For the interrupts to release low-power consumption mode, set the clock generator as well. 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.) o Setting to release low-power consumption mode Clock generator o Common NVIC registers	"8.5.3.2 Preparation"
Interrupt generation	An interrupt request is generated.	
Not clearing standby mode Clearing standby mode CG detects interrupt (clearing standby mode)	Interrupts used to release low-power consumption mode are connected to the CPU via the clock generator.	"8.5.3.3 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 based on the priority order.	"8.5.3.4 Detection by CPU"
CPU handles interrupt	The CPU handles the interrupt. The CPU preserves the contents of the register to the stack before branching to the ISR.	"8.5.3.5 CPU processing"

Processing	Details	See
ISR execution	Prepare the necessary ISRs. Clear the interrupt source if needed.	"8.5.3.6 Interrupt Service Routine (Clearing the sources)"
Return to preceding program	Return to the preceding normal program.	

8.5.3.2 Preparation

When preparing for an interrupt, make sure to follow the order of configuration to prevent an occurrence of unexpected interrupt on the way.

Initiating an interrupt or changing its configuration must be implemented in the following order: disable the interrupt by the CPU; and configure the settings starting the farthest route from the CPU. Then enable the interrupt by the CPU.

To configure the clock generator, follow the order indicated here not to cause an 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. Prohibit the interrupts to the CPU.
2. Configuring the interrupts to the CPU.
3. Pre configuration (1) (Interrupt from external interrupt pin)
4. Pre configuration (2) (Interrupt from peripheral function)
5. Pre configuration (3) (Interrupt Set-Pending Register)
6. Configuring the clock generator.
7. Permit the interrupts to the CPU.

(1) Prohibit the interrupts to the CPU

To prohibit the CPU from an interrupts, 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: The 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) Configure the interrupts to the CPU

A priority level can be set with <PRI_n> of the Interrupt Priority Register of the NVIC register.

Each interrupt source is provided with eight bits for assigning a priority level from 0 to 255; however, 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.

When a group priority is used, set <PRIGROUP> of the Application Interrupt and Reset Control Register.

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

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

(3) Pre configuration (1) (Interrupt from external interrupt pin)

When the external interrupt pins are used, set the port register of the corresponding pin. Setting PxIE[m] to "1" allows the pin to be used as the input port.

Port register		
PxIE<PxmIE>	←	"1"

Note: **x: port number / m: corresponding bit**
An interrupt input is enabled when PxIE is enabled. Do not enable the unused interrupts.

(4) Pre configuration (2) (Interrupt from peripheral function)

When the interrupts from the peripheral function are used, the interrupt setting varies depending on the peripheral function. See the chapter of each peripheral function for details.

(5) Pre configuration (3) (Interrupt Set-Pending Register)

When the interrupts are generated from 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

When the interrupts for releasing low power consumption mode are used, set the active level and interrupts of the CGIMCG register of the clock generator. The CGIMCG register is capable of configuring each source. According to the active level, refer to "Table 8-3 List of Interrupt Sources".

Before enabling an interrupt, clear unnecessary interrupt requests with the CGICRCG register. This can avoid unexpected interrupt. To clear corresponding interrupt request, write the value corresponding to the source to the CGICRCG register. See "8.6.3.2 CGICRCG (CG Interrupt Request Clear Register)" for each value.

When the interrupt requests from external pins is not used to release low-power consumption mode, the clock generator can be used without setting; however, an "High" level pulse or "High"-level signal must be input so that the CPU can detect it as an interrupt request.

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) Permit the interrupts to the CPU

Permit the interrupts to the CPU as described below:

Firstly, clear the suspended interrupt in the Interrupt Clear-Pending Register. Secondary, enable the interrupt with the Interrupt Set-Enable Register. Each bit of these registers is assigned to each 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.

Note that if interrupts are generated with the Interrupt Set-Pending Register setting, the sources for interrupts are cleared. Thus, Clearing suspended interrupt is not necessary.

Finally, set the PRIMASK register to "0".

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

Note 1: **m: corresponding bit**

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

8.5.3.3 Detection by Clock Generator

If an interrupt source is used for releasing low-power consumption mode, an interrupt request is detected according to the active level specified in the clock generator, and is notified to the CPU.

When an edge-triggered interrupt request is detected, the source is held in the clock generator. A level-sensitive interrupt request must keep its active level until it is detected, otherwise the interrupt request will be assumed to have been cleared when the signal level changes from active to inactive.

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

8.5.3.4 Detection by CPU

The CPU detects a highest priority interrupt request based on the priority order.

8.5.3.5 CPU processing

On detecting an interrupt, the CPU preserves the contents of xPSR, PC, LR, R12 and r3 to r0 to the stack, and then the CPU enters the ISR.

8.5.3.6 Interrupt Service Routine (Clearing the sources)

An ISR requires specific programming according to the application to be used. This section describes that the recommended service routine programming and how to clear the sources.

(1) Procedure during ISR

In the ISR, normally contents of the necessary register are preserved to the stack and interrupts are processed. The Cortex-M3 core automatically preserves the contents of xPSR, PC, LR, R12 and r3 to r0 to the stack. No extra programming is required for them.

For other registers, preserve the contents 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 preserve the contents of general-purpose registers because they might be rewritten.

(2) Clearing the interrupt sources

For the interrupt sources used for releasing low-power consumption mode, each interrupt request must be cleared with the CGICRCG register.

If an interrupt source is level-sensitive, the interrupt request continues to exist until the source is cleared. Therefore, the interrupt source should be cleared first. For detection of level-sensitive interrupts, if the interrupt source is cleared, the interrupt request signal from the clock generator is automatically cleared.

For detection of edge-sensitive interrupts, if the value corresponding to the interrupt is set to the CGICRCG register, the source is cleared. When an active edge occurs again, a new interrupt request will be detected.

8.6 Exception/Interrupt-Related Registers

8.6.1 List of Registers

The following table shows the control registers and their addresses.

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

NVIC registers Base Address = 0xE000_E000

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

Peripheral function name: CG

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

Note: Do not access to the areas described as "Reserved".

8.6.2 NVIC Registers

8.6.2.1 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 has been 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) 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> <HOSCON> by 32 is used as external reference clock.

8.6.2.2 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".

8.6.2.3 SysTick Current Value Register

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

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

8.6.2.4 SysTick Calibration Value Register

	31	30	29	28	27	26	25	24
Bit symbol	NOREF	SKEW	-	-	-	-	-	-
After reset	0	1	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	1	0	0
	7	6	5	4	3	2	1	0
Bit symbol	TENMS							
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31	NOREF	R	0: Reference clock provided 1: No reference clock
30	SKEW	R	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.
29-24	−	R	Read as "0".
23-0	TENMS	R	Calibration value This calibration value is a reloaded value (0xC35) count a 10 ms using the external reference clock. (Note)

Note: When a multi-shot timer is used, use this value of N-1.

8.6.2.5 Interrupt Control Registers

Each interrupt source has the interrupt set-enable register, interrupt clear-enable register, interrupt set-pending register, and interrupt clear-pending register.

Each bit corresponds to the specified interrupt.

(1) Interrupt Set-Enable Register

This register enables interrupts and checks whether the interrupt is enabled or disabled.

When set this register to "1", the corresponding interrupt is enabled.

Writing "0" has no meaning.

Reading this register can check whether the corresponding interrupt is enabled or disabled.

To clear the bit of this register, the corresponding bit of interrupt clear-enable register is cleared to "0".

Bit symbol	Type	Function
SETENA	R/W	Interrupt No. [31:0] [Write] 1: Enable interrupt [Read] 0: The interrupt is disabled. 1: The interrupt is enabled.

Note: For each description of the interrupt and the interrupt numbers, refer to "8.5.2 List of Interrupt Sources".

(a) Interrupt Set-Enable Register

	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 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 1)	SETENA (Interrupt 0)
After reset	0	0	0	0	0	0	0	0

(2) Interrupt Clear-Enable Register

This register disables interrupts and checks whether the interrupt is enabled or disabled.

When set this register to "1", the corresponding interrupt is disabled.

Writing "0" has no meaning.

Reading this register can check whether the corresponding interrupt is enabled or disabled.

Bit symbol	Type	Function
CLRENA	R/W	Interrupt No. [31:0] [Write] 1: Disable interrupt [Read] 0: The interrupt is disabled 1: The interrupt is enabled

Note: For each description of the interrupt and the interrupt numbers, refer to "8.5.2 List of Interrupt Sources".

(a) Interrupt Clear-Enable Register

	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 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 1)	CLRENA (Interrupt 0)
After reset	0	0	0	0	0	0	0	0

(3) Interrupt Set-Pending Register

This register forcibly pends interrupts and checks whether the interrupt is pended.

When "1" is set to this register, the corresponding interrupt is pended. However, this register is invalid for the interrupt which has been already pended or disabled.

Writing "0" has no meaning.

Reading this register can check whether the corresponding interrupt is pending.

To clear the bit of this register, set "1" to the corresponding bit of interrupt clear-pending register.

Bit symbol	Type	Function
SETPEND	R/W	Interrupt No. [31:0] [Write] 1: Pends an interrupt. [Read] 0: No pending interrupt. 1: A pending interrupt exists.

Note: For each description of the interrupt and the interrupt numbers, refer to "8.5.2 List of Interrupt Sources".

(a) Interrupt Set-Pending Register

	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 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 1)	SETPEND (Interrupt 0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

(4) Interrupt Clear-Pending Register

This register clears the pending interrupts and checks whether the interrupt is pending.

When "1" is set to this register, the corresponding pending interrupt is cleared. However, this register is invalid for the interrupt which has been already started.

Writing "0" has no meaning.

Reading this register can check whether the corresponding interrupt is pending.

Bit symbol	Type	Function
SETPEND	R/W	Interrupt No. [31:0] [Write] 1: Clears a pending interrupt. [Read] 0: No pending interrupt 1: A pending interrupt exists.

Note: For each description of the interrupt and the interrupt numbers, refer to "8.5.2 List of Interrupt Sources".

(a) 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 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 1)	CLRPEND (Interrupt 0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

8.6.2.6 Interrupt Priority Register

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

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

	31 24	23 16	15 8	7 0
0xE000_E400	-	-	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_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28

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

The following table shows the fields of the Interrupt Priority Registers for interrupt numbers 0 to 3 as a representative example. Reading unused bits return "0", and writing data 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".

8.6.2.7 Vector Table Offset Register

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

Bit	Bit symbol	Type	Function
31-7	TBLOFF	R/W	Offset value Set the offset value from the address 0x0000_0000. 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, adjust the alignment by rounding up to the next power of two.
6-0	−	R	Read as "0".

8.6.2.8 Application Interrupt and Reset Control Register

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

Bit	Bit symbol	Type	Function
31-16	VECTKEY (Write)/ VECTKEY-STAT(Read)	R/W	Register key [Write] Writing to this register requires writing "0x5FA" to the <VECTKEY> field. [Read] Read as "0xFA05".
15	ENDIANESS	R/W	Endianness bit: (Note1) 1: Big endian 0: Little endian
14-11	−	R	Read as "0".
10-8	PRIGROUP	R/W	Interrupt priority grouping 000: Seven bits of pre-emption priority, one bit of sub-priority 001: Six bits of pre-emption priority, two bits of sub-priority 010: Five bits of pre-emption priority, three bits of sub-priority 011: Four bits of pre-emption priority, four bits of sub-priority 100: Three bits of pre-emption priority, five bits of sub-priority 101: Two bits of pre-emption priority, six bits of sub-priority 110: One bit of pre-emption priority, seven bits of sub-priority 111: No pre-emption priority, eight bits of sub-priority The bit configuration of the interrupt priority register <PRI_n> can be split into pre-emption priority and sub priority.
7-3	−	R	Read as "0".
2	SYSRESET REQ	R/W	System Reset Request. When "1" is set to this bit, the CPU outputs a SYSRESETREQ signal. (Note 2)
1	VECTCLR ACTIVE	R/W	Clear an active vector bit. 1: Clears all state information of active NMI, fault, and interrupts 0: Not cleared. This bit is cleared by the self operation. Stack initializing should be performed by the application.
0	VECTRESET	R/W	System Reset bit 1: Rests the system 0: No system reset. When "1" is set to this bit, the internal systems of the CPU except the debug components (FPB, DWT and ITM) are reset. This bit is also zero cleared.

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

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

8.6.2.9 System Handler Priority Register

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

The following table 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 table shows the fields of the System Handler Priority Registers for interrupt numbers 4 to 7 as a representative example. Reading unused bits return "0", and writing data 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".

8.6.2.10 System Handler 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	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	SVCALL PENDED	BUSFAULT PENDED	MEMFAULT PENDED	USGFAULT PENDED	SYSTICKACT	PENDSVACT	-	MONITOR ACT
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
After reset	0	0	0	0	0	0	0	0

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

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

Note: **Be careful to rewrite the active bits with extreme caution because the contents of the stack is not updated.**

8.6.3 Clock Generator Registers

8.6.3.1 CGIMCGA (CG Interrupt Mode Control Register)

The CG interrupt mode control register specifies the active level to release the low power consumption mode and enables or disables the release of the low power consumption mode. Detected active level can be read from this register.

Bit symbol	Type	Function
EMCGx[2:0]	R/W	Select the active level to release the low power consumption mode. The active level can be selected from Table 8-4. 000: "Low" level 001: "High" level 010: Falling edge 011: Rising edge 100: Both edges Values other than the above are prohibited.
EMSTx[1:0]	R	Detected active level (This bit is valid only when EMCGx[2:0]="100".) 00: − 01: Rising edge 10: Falling edge 11: Both edges
INTxEN	R/W	Releases the low power consumption mode. 0: Disabled 1: Enabled

Table 8-4 The Active Level to release the Low Power Consumption Mode

Interrupt Source		Active level control register	The active level to release the low power consumption mode				
			"Low" level	"High" level	Rising edge	Falling edge	Both edge
INT6	External interrupt pin 6	CGIMCGA <EMCG0[2:0]>	o	o	o	o	o
INT7	External interrupt pin 7	CGIMCGA <EMCG1[2:0]>	o	o	o	o	o
INTC	External interrupt pin C	CGIMCGA <EMCG2[2:0]>	o	o	o	o	o

Note: The active level marked with "o" can be used to release the low power consumption mode. The active level marked with "x" cannot be used.

(1) CGIMCGA (CG Interrupt Mode Control Register A)

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

Note 1: The active level specified with <EMCGx[2:0]> depends on the interrupt request. Refer to Table 8-4.

Note 2: **<EMSTx> is valid only when <EMCGx[2:0]> is set to "100" for both rising and falling edge. In other cases, it becomes undefined. The active level used for releasing low-power consumption mode can be checked by referring <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.**

Note 3: Do not specify <INTxEN> simultaneously with the edge. Set the edge first and then specify <INTxEN>.

Note 4: "0" is read from bit 31 to bit24.

Note 5: "0" is read from bit 23, 15 and 7.

Note 6: Undefined value is read from bit 17, 9 and 1.

8.6.3.2 CGICRCG (CG Interrupt Request Clear Register)

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

Bit	Bit symbol	Type	Function
31-5	−	R	Read as "0".
4-0	ICRCG[4:0]	W	Clear interrupt requests. 0_0000: INT6 0_0001: INT7 0_0010: INTC 0_0011 to 1_1111: Prohibited Read as "0".

8.6.3.3 CGNMIFLG (NMI Flag Register)

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

Bit	Bit symbol	Type	Function
31-1	−	R	Read as "0".
0	NMIFLG0	R	NMI source generation flag 0: No flag. 1: NMI is generated from the WDT.

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

8.6.3.4 CGRSTFLG (Reset Flag Register)

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

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

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

Note 2: This product has the 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 immediately 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.

9. Digital Noise Filter Circuit (DNF)

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

9.1 Configuration

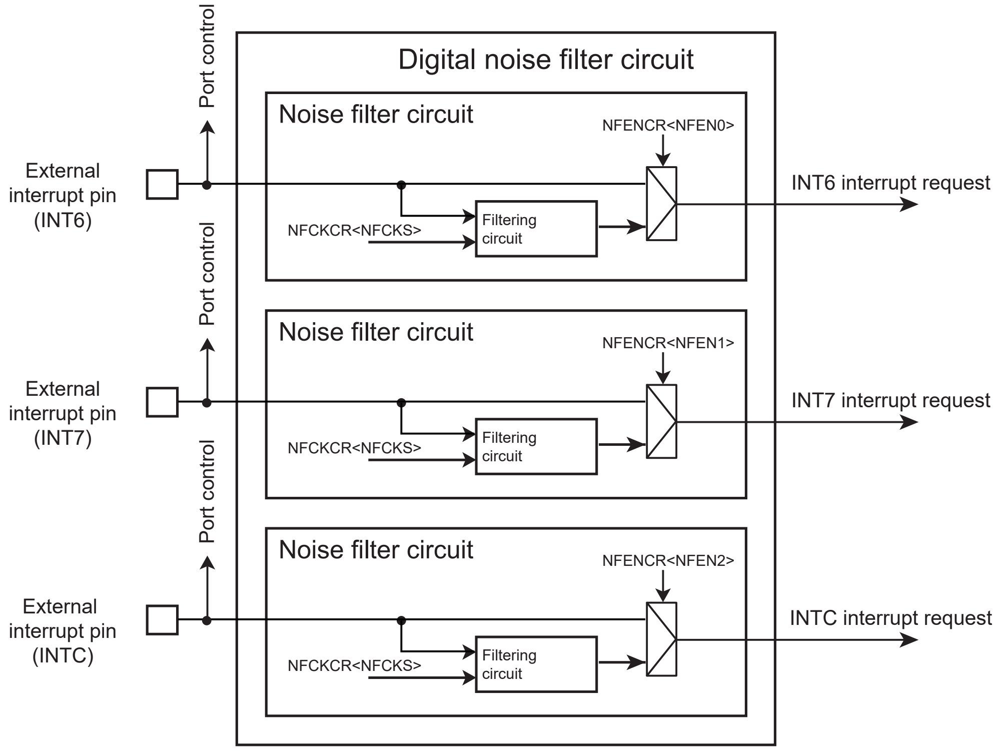


Figure 9-1 Circuit diagram of digital noise filter

9.2 Registers

9.2.1 Register List

Base Address = 0x4006_0000

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

9.2.1.1 NFCKCR (Noise Filter Control Register)

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

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

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

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

9.2.1.2 NFENCR (Noise Filter Enable register)

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

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

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

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

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

9.3 Operation Description

9.3.1 Configuration

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

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

9.3.2 Operation

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

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

9.3.3 Noise Filter Usable Operation Mode

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

9.3.4 Precautions on Use of STOP Mode

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

9.3.5 Minimum Noise Filtering Time

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

Table 9-1 Minimum noise filtering time

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

10. Input / Output Port

This chapter describes port-related registers, their settings and circuits.

10.1 Registers

When the port registers are used, the following registers must be set.

All registers are 32-bits. The configurations are different depending on the number of port bits and assignation of the function.

"x" means the name of ports and "n" means the function number in the following description.

Register Name		Setting Value	
PxDATA	Data register	0 or 1	This register reads / writes port data.
PxCR	Output control register	0: Output Disable 1: Output Enable	This register controls output.
PxFRn	Function register n	0: PORT 1: Function	This register sets the function. The assigned function can be enabled by setting "1". This register exists for the each function assigned to the port. In case of having some function, only one function can be enabled.
PxOD	Open-drain control register	0: CMOS 1: Open-drain	This register controls programmable open-drain outputs. Programmable open-drain outputs are set with PxOD. When output data is "1", output buffer is disabled and becomes a pseudo-open-drain output.
PxPUP	Pull-up control register	0: Pull-up Disable 1: Pull-up Enable	This register controls programmable pull-ups.
PxPDN	Pull-down control register	0: Pull-down Disable 1: Pull-down Enable	This register controls programmable pull-downs.
PxIE	Input control register	0: Input Disable 1: Input Enable	This register controls inputs. Some time is required after enabling PxIE until external data is reflected in PxDATA.

10.1.1 Register List

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

Register name	Address (Base+)	PORT B	PORT E	PORT F	PORT J	PORT K
Data register	0x0000	PBDATA	PEDATA	PFDATA	PJDATA	PKDATA
Output control register	0x0004	PBCR	PECR	PFCR	PJCR	PKCR
Function register 1	0x0008	PBFR1	PEFR1	PFFR1	−	−
Function register 2	0x000C	PBFR2	PEFR2	PFFR2	−	−
Function register 3	0x0010	−	−	−	−	−
Function register 4	0x0014	PBFR4	PEFR4	PFFR4	−	−
Function register 5	0x0018	−	PEFR5	−	−	−
Function register 6	0x001C	−	PEFR6	−	−	−
Open-drain control register	0x0028	PBOD	PEOD	PFOD	PJOD	PKOD
Pull-up control register	0x002C	PBPUP	PEPUP	PFPUP	PJPUP	PKPUP
Pull-down control register	0x0030	PBPDN	PEPDN	PFPDN	PJPDN	PKPDN
Input control register	0x0038	PBIE	PEIE	PFIE	PJIE	PKIE

Register name	Address (Base+)	PORT M
Data register	0x0000	PMDATA
Output control register	0x0004	PMCR
Function register 1	0x0008	−
Function register 2	0x000C	−
Function register 3	0x0010	−
Function register 4	0x0014	−
Function register 5	0x0018	−
Function register 6	0x001C	−
Open-drain control register	0x0028	PMOD
Pull-up control register	0x002C	PMPUP
Pull-down control register	0x0030	PMPDN
Input control register	0x0038	PMIE

Note: Do not access to the address shown as "-".

10.1.2 Port Function and Setting List

The list of the function and setting register for each port is as shown below:

- "Table 10-1 PORT B Setting List"
- "Table 10-2 PORT E Setting List"
- "Table 10-3 PORT F Setting List"
- "Table 10-4 PORT J Setting List"
- "Table 10-5 PORT K Setting List"
- "Table 10-6 PORT M Setting List"

The cell of PxFRn shows the function register which must be set to select a function. If this register is set to "1", the corresponding function is enabled.

A bit in the cell filled with a hatch is read as "0" and the writing a data to this bit is invalid.

"0" or "1" in the table is shown the value which is set to the register. "0/1" is shown that the optional value can be set to the register.

Some function input / output pins are assigned to some ports. Only one port can be assigned by function registers.

Pxm: P (port) + port name "x" and its register bit "m". For example, PB0 represents that the port name B and bit 0 of its register.

10.1.2.1 PORT B

Table 10-1 PORT B Setting List

PO RT	Reset status	Input/ Output	PORT Type	Control registers						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB3	After reset (SWDIO)			0	1	PBFR1	0	1	0	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SWDIO	I/O	FT2	0/1	1	PBFR1	0/1	0/1	0/1	1
	TB0OUT	Output	FT1	0/1	1	PBFR2	0/1	0/1	0/1	0
	(SB0SDA)	I/O	FT1	0/1	1	PBFR4	1	0/1	0/1	1
PB4	After reset (SWCLK)			0	0	PBFR1	0	0	1	1
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SWCLK	Input	FT2	0/1	0	PBFR1	0/1	0/1	0/1	1
	TB0IN	Input	FT1	0/1	0	PBFR2	0/1	0/1	0/1	1
	(SB0SCL)	I/O	FT1	0/1	1	PBFR4	1	0/1	0/1	1

10.1.2.2 PORT E

Table 10-2 PORT E Setting List

PORT	Reset status	Input/ Output	PORT Type	Control registers						
				PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SC0TXD	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
	SB0SCL	I/O	FT1	0/1	1	PBFR2	1	0/1	0/1	1
	INT6	Input	FT4	0/1	0	PEFR4 (Note)	0/1	0/1	0/1	1
	TB7OUT	Output	FT1	0/1	1	PEFR5	0/1	0/1	0/1	0
PE1	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SC0RXD	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
	TB4IN	Input	FT1	0/1	0	PEFR2	0/1	0/1	0/1	1
PE2	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	SC0SCLK	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
		Output		0/1	1	PEFR1	0/1	0/1	0/1	0
	$\overline{\text{SC0CTS}}$	Input	FT1	0/1	0	PEFR2	0/1	0/1	0/1	1
	INT7	Input	FT4	0/1	0	PEFR4 (Note)	0/1	0/1	0/1	1
	TB5OUT	Output	FT1	0/1	1	PEFR6	0/1	0/1	0/1	0

Note: When external interrupts are used to release STOP mode, set "1" to corresponding bit of PEFR4; when external interrupts are not used to release STOP mode, set "0" to the corresponding bit of PEFR4.

10.1.2.3 PORT F

Table 10-3 PORT F Setting List

PO	Reset status	Input/	PORT	Control registers						
RT		Output	Type	PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF0	After reset			0	0	0	0	0	0	0
	Input Port	Input		0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output		0/1	1	0	0/1	0/1	0/1	0
	TB7IN	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SB0SDA	I/O	FT1	0/1	1	PFFR2	1	0/1	0/1	1
	INTC	Input	FT4	0/1	0	PFFR4 (Note 2)	0/1	0/1	0/1	1

Note 1: PF0 works as a BOOT function. It is enabled to be input and pulled-up while RESET pin is "Low". At the rising edge of the reset signal, if PF0 is "High", the device enters single chip mode and boots from the on-chip flash memory. If PF0 is "Low", the device enters single BOOT mode and boots from the internal BOOT program.

Note 2: When external interrupts are used to release STOP mode, set "1" to corresponding bit of PEFR4; when external interrupts are not used to release STOP mode, set "0" to the corresponding bit of PEFR4.

10.1.2.4 PORT J

Table 10-4 PORT J Setting List

PO	Reset status	Input/	PORT	Control registers						
RT		Output	Type	PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ5	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AINB8	Input	FT5	0/1	0		0/1	0	0	0
PJ6	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AINB9	Input	FT5	0/1	0		0/1	0	0	0
PJ7	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AINB10	Input	FT5	0/1	0		0/1	0	0	0

10.1.2.5 PORT K

Table 10-5 PORT K Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PKDATA	PKCR	PKFRn	PKOD	PKPUP	PKPDN	PKIE
PK0	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AINB11	Input	FT5	0/1	0		0/1	0	0	0
PK1	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	AINB12AINB16	Input	FT5	0/1	0		0/1	0	0	0

10.1.2.6 PORT M

Table 10-6 PORT M Setting List

PO RT	Reset status	Input/Output	PORT Type	Control registers						
				PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE
PM0	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	X1	Input		0/1	0		0/1	0/1	0/1	1
PM1	After reset			0	0		0	0	0	0
	Input Port	Input		0/1	0		0/1	0/1	0/1	1
	Output Port	Output		0/1	1		0/1	0/1	0/1	0
	X2	Input		0/1	0		0/1	0/1	0/1	1

10.2 Block Diagrams of Ports

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".

The operation of "Direct reset" shown in the circuit diagram is enabled when the cold-reset occurs or when the STOP2 mode is released by the reset pin.

10.2.1 Type FT1

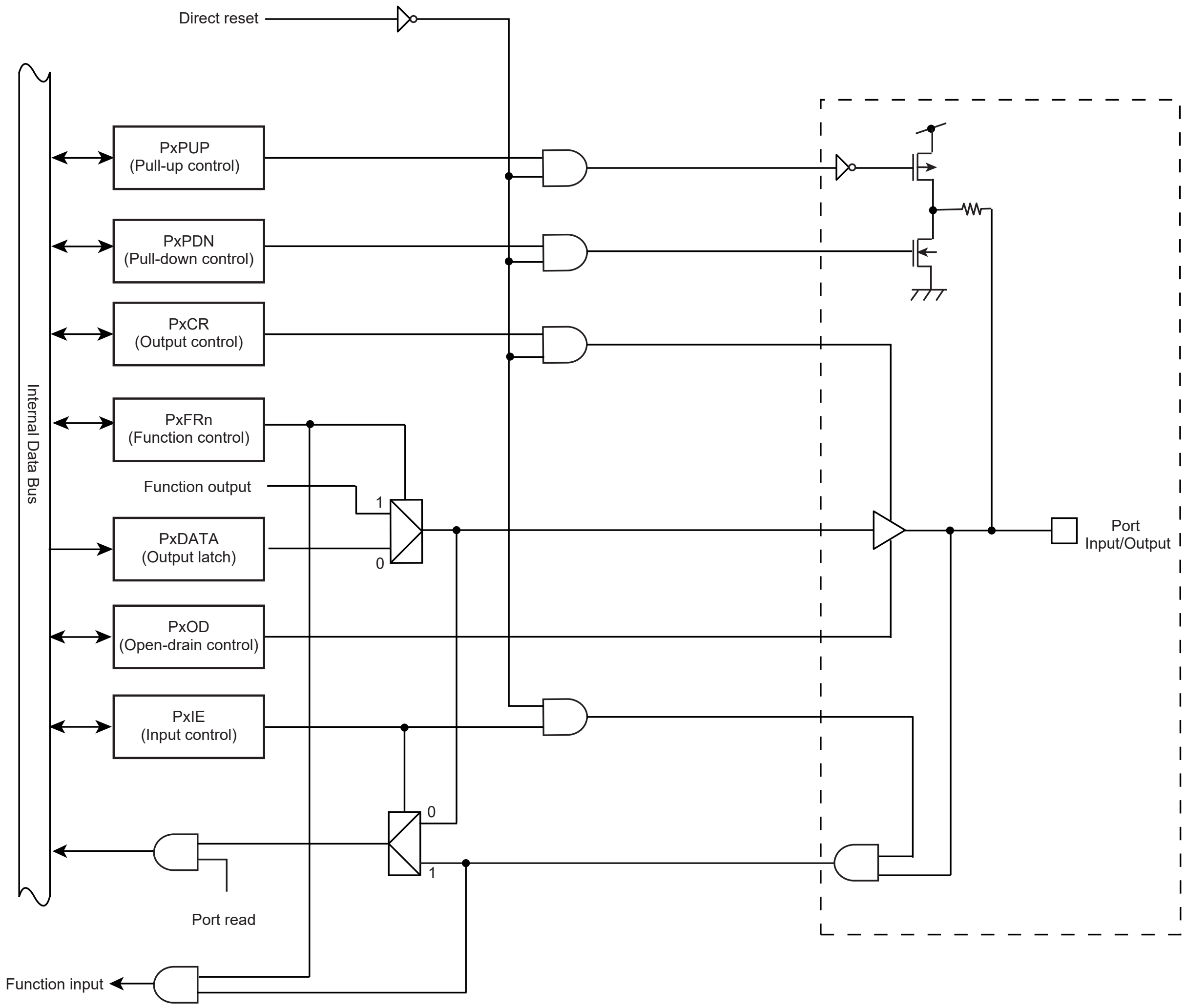


Figure 10-1 Port Type FT1

10.2.2 Type FT2

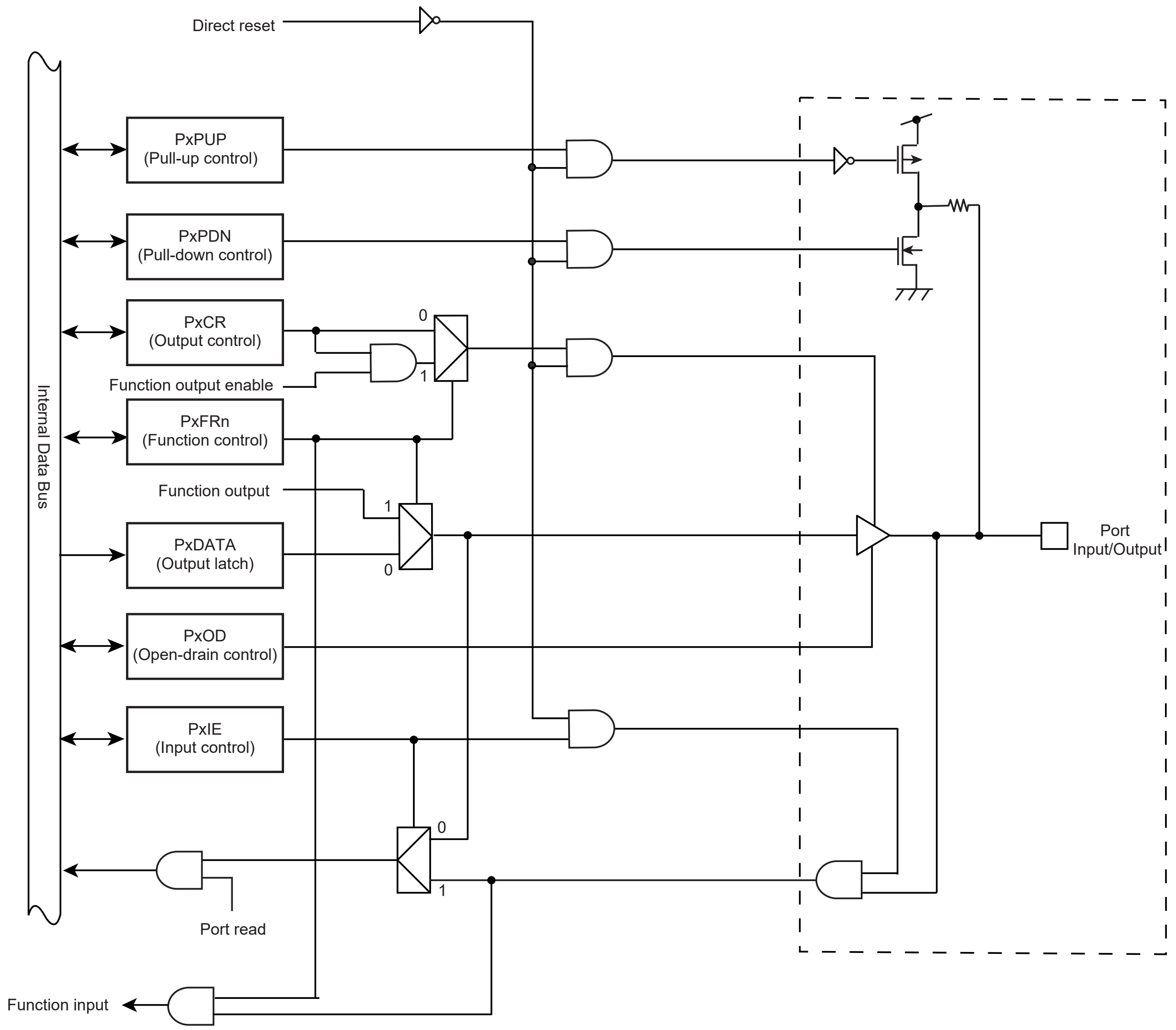


Figure 10-2 Port Type FT2

10.2.3 Type FT3

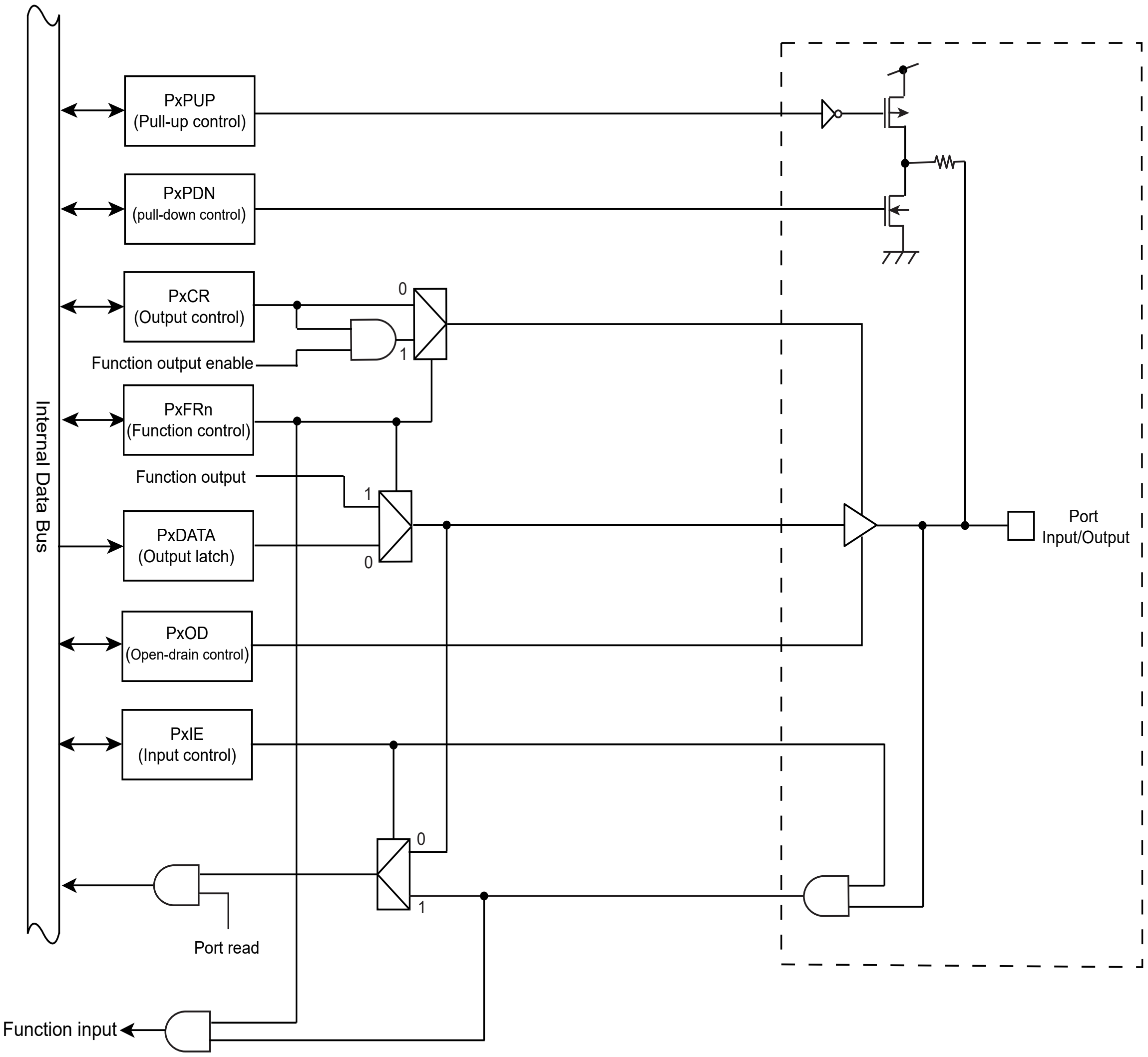


Figure 10-3 Port Type FT3

10.2.4 Type FT4

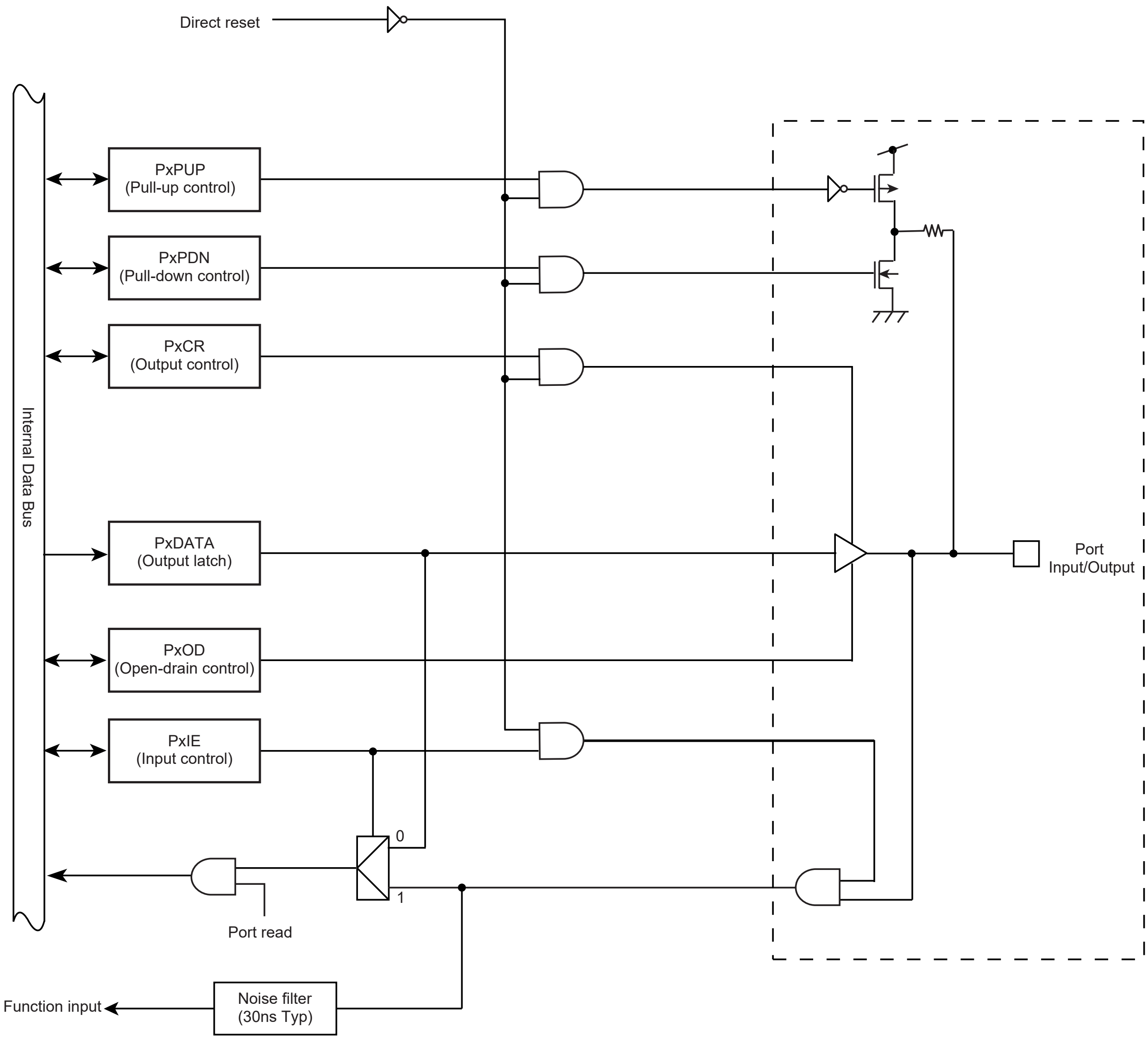


Figure 10-4 Port Type FT4

10.2.5 Type FT5

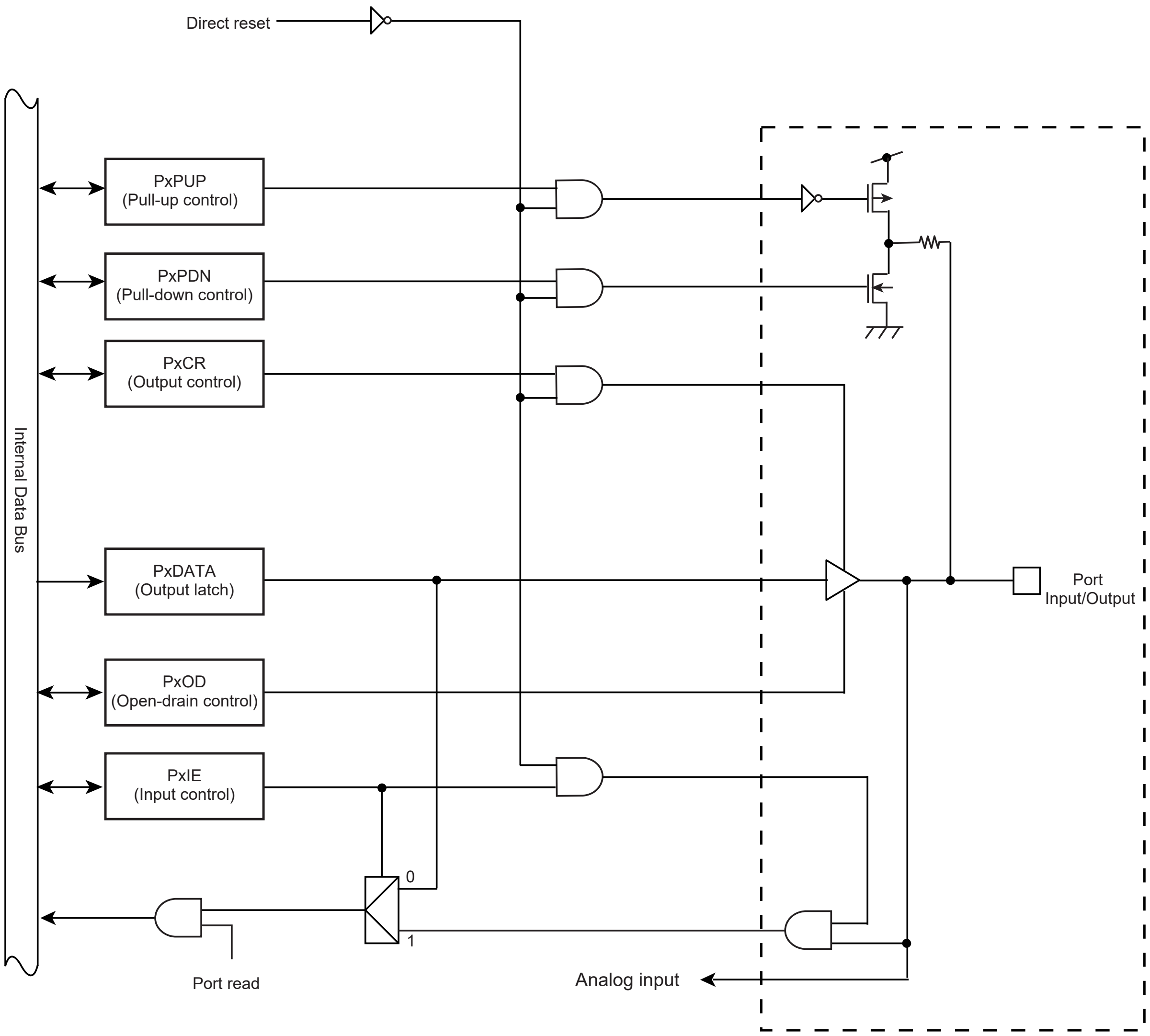


Figure 10-5 Port Type FT5

10.2.6 Type FT6

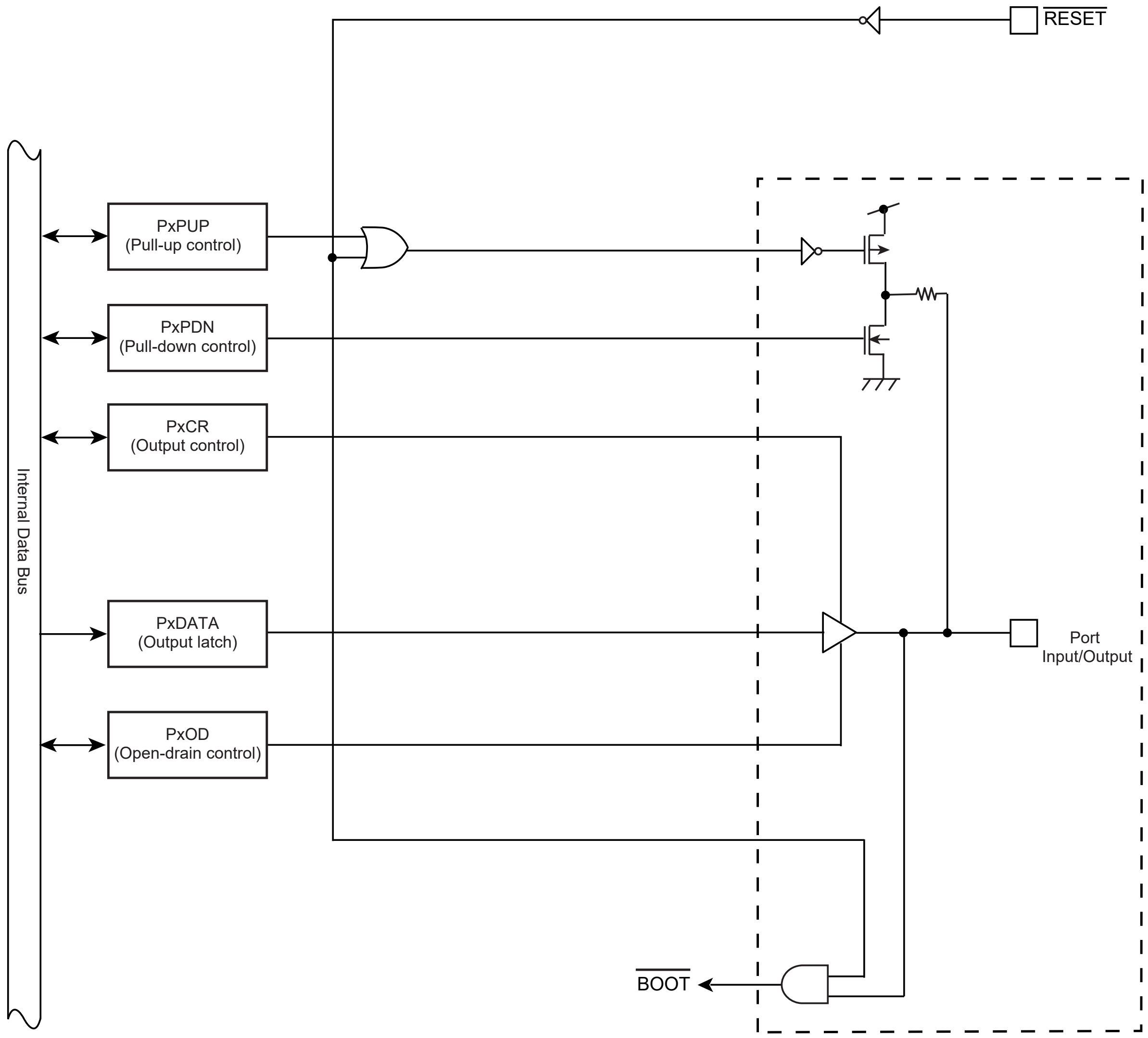


Figure 10-6 Port Type FT6

10.2.7 Type FT7

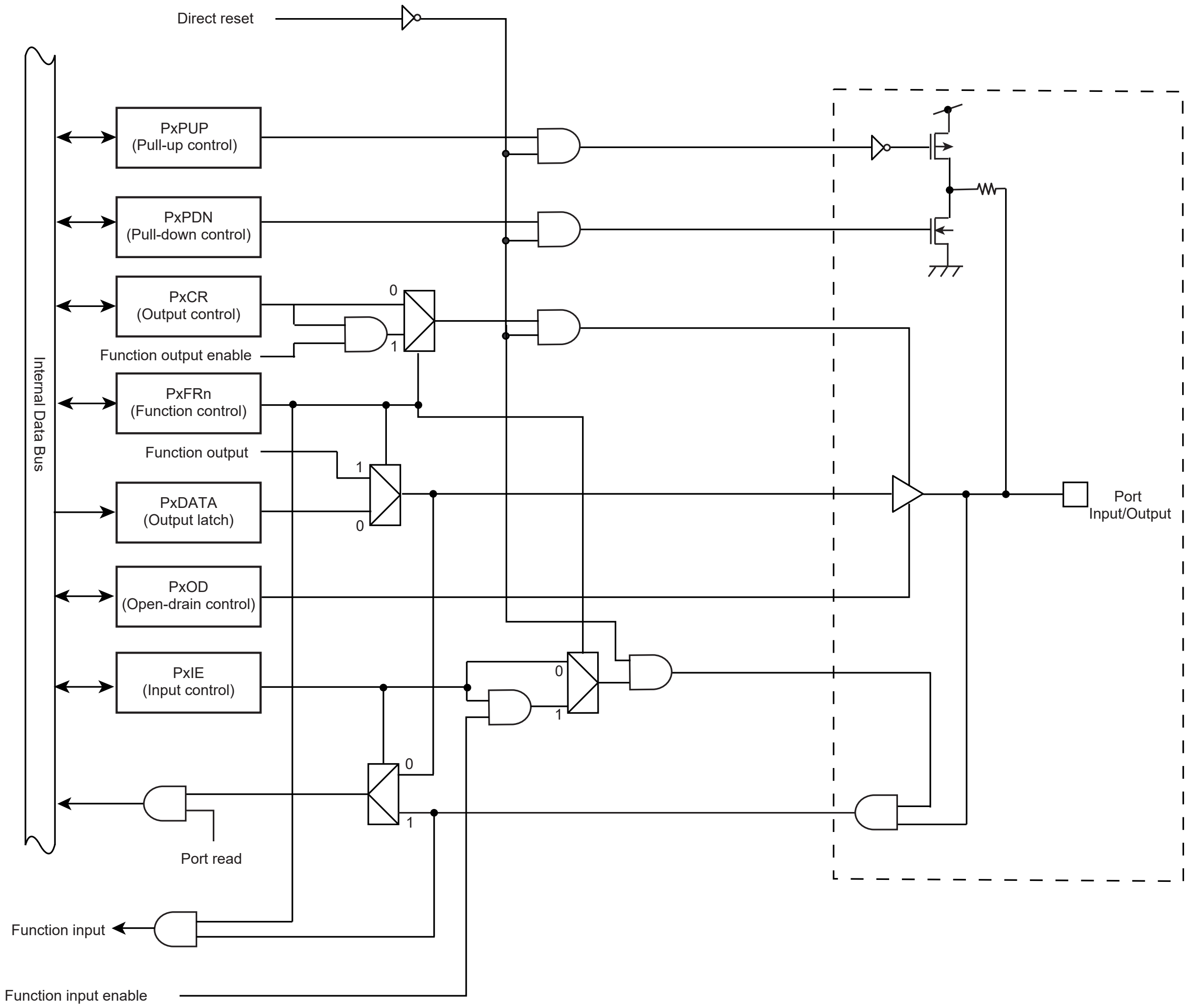


Figure 10-7 Port Type FT7

10.2.8 Type FT8

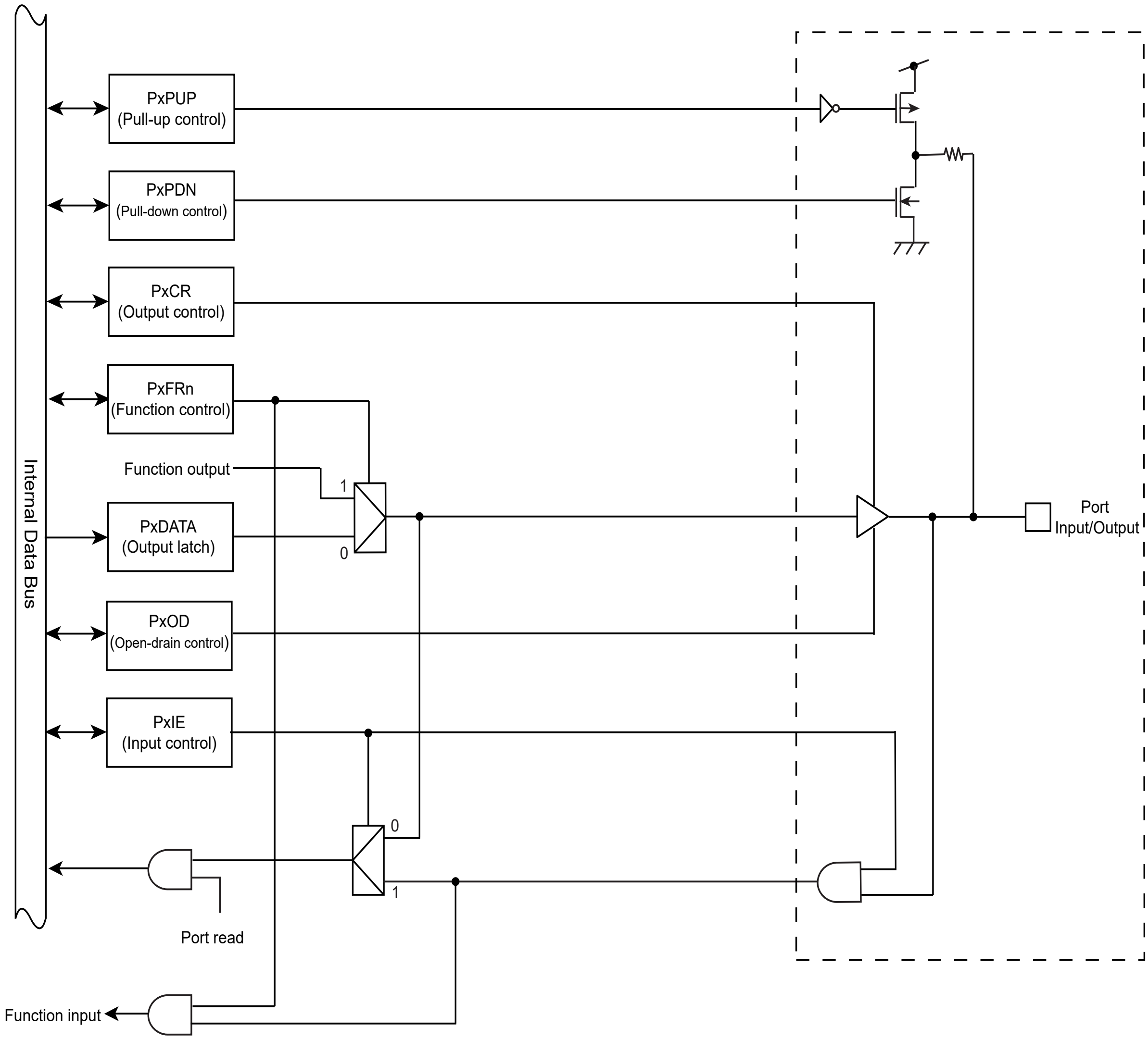


Figure 10-8 Port Type FT8

11. 16-bit Timer / Event Counters (TMRB)

11.1 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.

11.2 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.

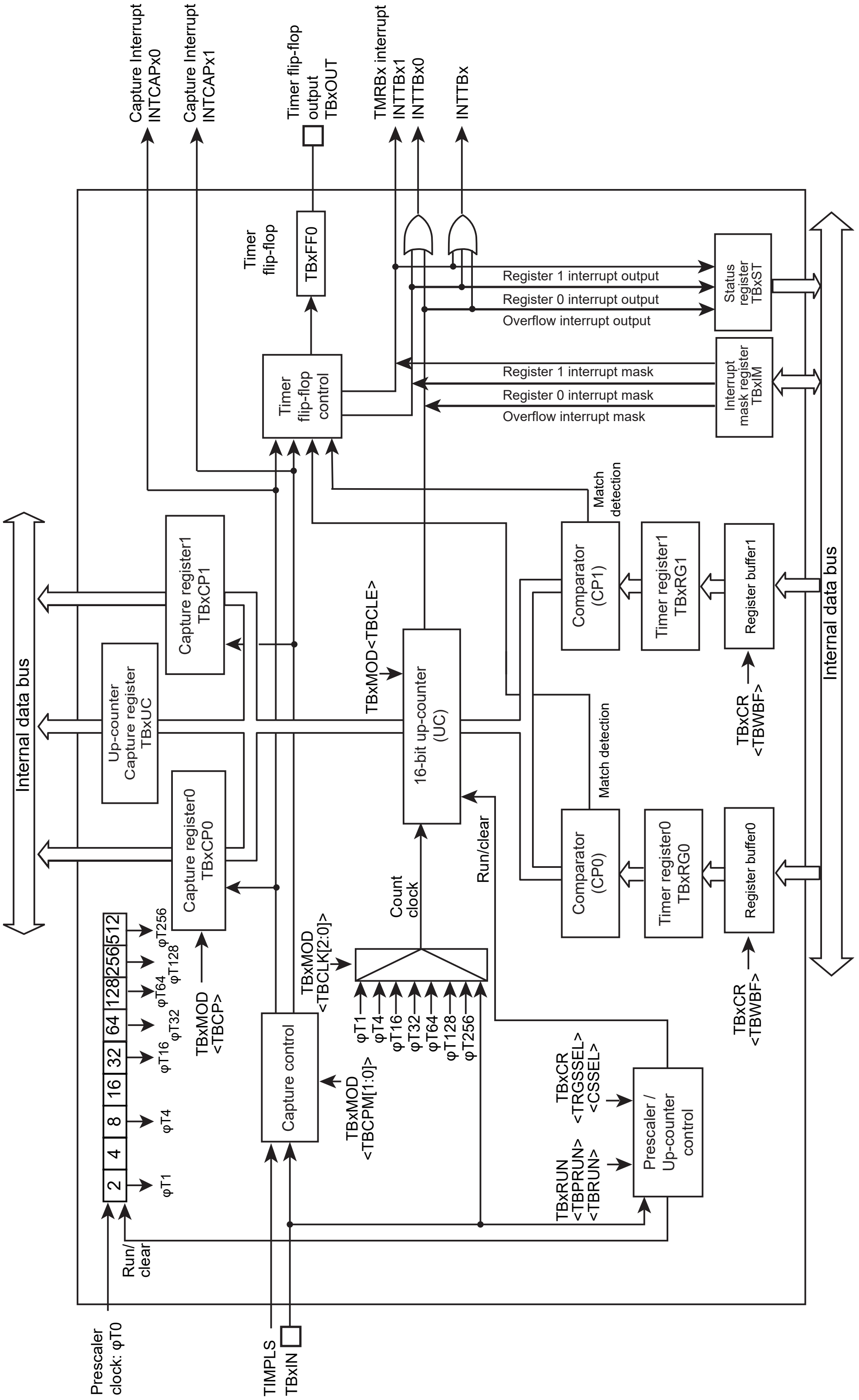


Figure 11-1 TMRBx Block Diagram

11.3 Registers

11.3.1 Register list according to channel

The table below shows control registers and their addresses.

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

Register name		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

11.3.2 TBxEN(Enable register)

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

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

11.3.3 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.

11.3.4 TBxCR(Control register)

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

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

Note 1: Do not modify TBxCR during operating TMRB.

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

11.3.5 TBxMOD(Mode register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBRSWR	R/W	Writes to timer registers 0 and 1 (when double buffering is enabled) 0: The data transfer to the timer register 0 and 1 is done by corresponding to the up-counter (UC) regardless of the rewriting of the buffer register 0 and 1. 1: To transfer the buffer registers data to the timer registers, the writing of the timer register 0 and 1 together are needed.
6	TBCP	W	Capture control by software 0: Capture by softwareTB 1: Don't care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as "1".
5-4	TBCPM[1:0]	R/W	Capture timing 00: Disable Capture timing 01: TBxIN↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. 10: TBxIN↑ TBxIN↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN pin input. 11: TIMPLS↑ TIMPLS↓ Takes count values into capture register 0 (TBxCP0) upon rising of TIMPLS input. Takes count values into capture register 1 (TBxCP1) upon falling of TIMPLS input.
3	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up counter when there is a match with Timer Regsiter1 (TBxRG1).
2-0	TBCLK[2:0]	R/W	Selects the TMRBx source clock. 00: TBxIN pin input 001: φT1 010: φT4 011: φT16 100: φT32 101: φT64 110: φT128 111: φT256

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

11.3.6 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.

11.3.7 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.

11.3.8 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.

11.3.9 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 captured value of the last time read TBxUC is read, and the current value of the up-counter is captured.

11.3.10 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.

11.3.11 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.

11.3.12 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.

11.3.13 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.

11.4 Description of Operations for Each Circuit

11.4.1 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.

11.4.2 Up-counter (UC)

UC is a 16-bit binary counter.

11.4.2.1 Source clock

The up-counter's source clock is specified by TBxMOD<TBCLK[2:0]>.

It can be selected from prescalere output clock - φT1, φT4, φT16 ,φT32, φT64, φT128 and φT256 - or the external clock of the TBxIN pin.

11.4.2.2 Counter start / stop

There are software start, external trigger start and synchronous start to start the counter.

1. Software start

   If TBxRUN<TBRUN> is set to "1", the counter will start. If "0" is set to the <TBRUN>, the counter will stop and the up-counter will be cleared at the same time.

2. External trigger start

   In the external trigger mode, the counter will be started by external signals.

   If TBxCR<CSSEL> is set to "1", the external trigger start mode is set. At this time, if <TBRUN> is set to "1", the condition of the counter will be trigger wait. The counter will start on the rising/falling edge of TBxIN0TBxIN.

   TBxCR<TRGSEL> bit specifies the switching external trigger edges.

   - <TRGSEL> = "0": Rising edge of TBxIN is selected.
   - <TRGSEL> = "1": Falling edge of TBxIN is selected.

   If <TBRUN> is set to "0", the counter will stop and the up-counter will be cleared at the same time.

3. Synchronous start

   In the timer synchronous mode, synchronous start timers can be possible. If timer synchronous mode is used in the PPG output mode, motor drive application can be achieved.

   Depending on products, the combination of master channels and slave channels have already been determined. For the combination of master channels and slave channels of this product, refer to Chapter Product Information.

TBxCR<TBSYNC> bit specifies the switching of synchronous mode. If <TBSYNC> bit of a slave channel is set to "1", the counter will start/stop synchronously with the software or external trigger start of a master channel. TBxRUN<TBPRUN, TBRUN> bit of a slave channel is not required to set. <TBSYNC> bit of the master channel must be set to "0".

Note that if the external trigger counter mode and timer synchronous mode are both set, the timer synchronous mode gains a higher priority.

11.4.2.3 Counter Clear

The up-counter is cleared at the timings below:

1. When a match with TBxRG1 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 up-counter stops

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

11.4.2.4 Up-Counter Overflow

If up-counter overflows, the INTTBx overflow interrupt is generated.

11.4.3 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.

11.4.4 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>.

11.4.5 Capture register (TBxCP0, TBxCP1)

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

11.4.6 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.

11.4.7 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.

11.4.8 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.

11.4.9 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.

11.5 Description of Operations for Each Mode

11.5.1 16-bit Interval Timer Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits INTTBx1 interrupt by setting corresponding bit to "1".
TBxFFCR	←	X	X	0	0	0	0	1	1	Disable to TBxFF0 reverse trigger.
TBxMOD	←	0	1	0	0	1	*	*	*	Changes to prescaler output clock as input clock. Specifies capture function to disable.
		(*** = 001, 010, 011,100,101,110,111)								
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a time interval. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don’t care −; No change

11.5.2 16-bit Event Counter Mode

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

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

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

Note: X; Don’t care −; No change

11.5.3 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

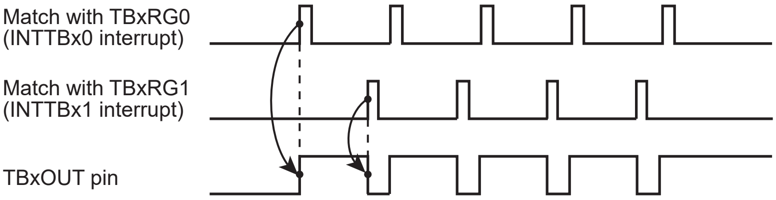


Figure 11-2 Example of Output of Programmable Pulse Generation (PPG)

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.

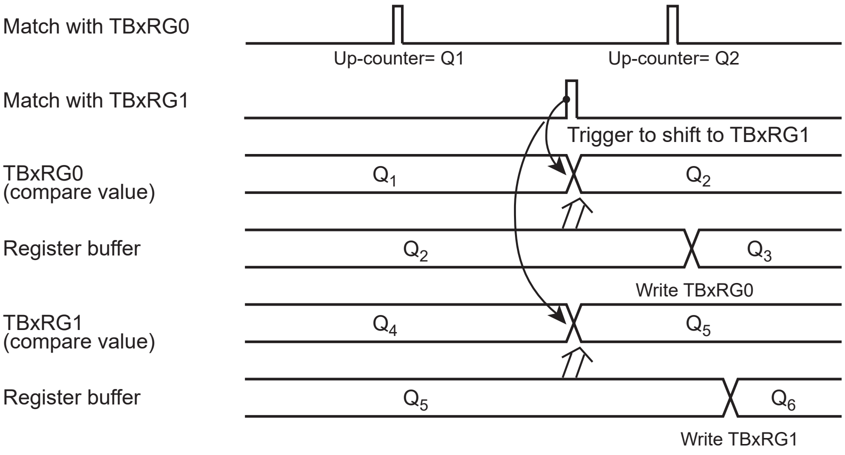


Figure 11-3 Register Buffer Operation

The block diagram of this mode is shown below.

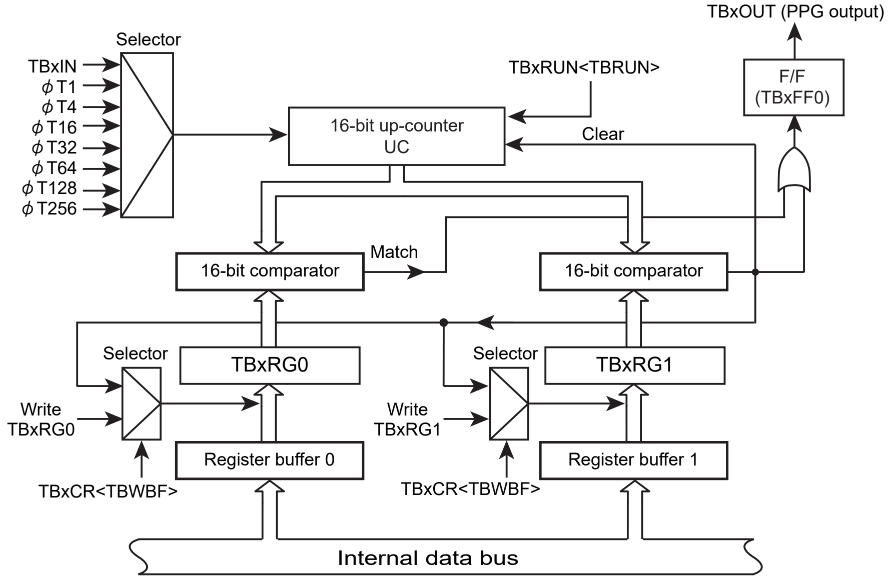


Figure 11-4 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxCR	←	0	0	0	X	−	0	0	0	Disables double buffering.
TBxRG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxRG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
	←	*	*	*	*	*	*	*	*	
TBxCR	←	1	0	0	X	−	0	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	←	0	1	0	0	1	*	*	*	Designates the prescaler output clock as the input clock,and disables the capture function.
		(** = 001, 010, 011,100,101,110,111)								
										UC is cleared to match TBxRG1.
Set PORT registers.										Allocates corresponding port to TBxOUT.
TBxRUN	←	*	*	*	*	*	1	X	1	Starts TMRBx.

Note: X; Don’t care
−; No change

11.5.4 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 11-5.

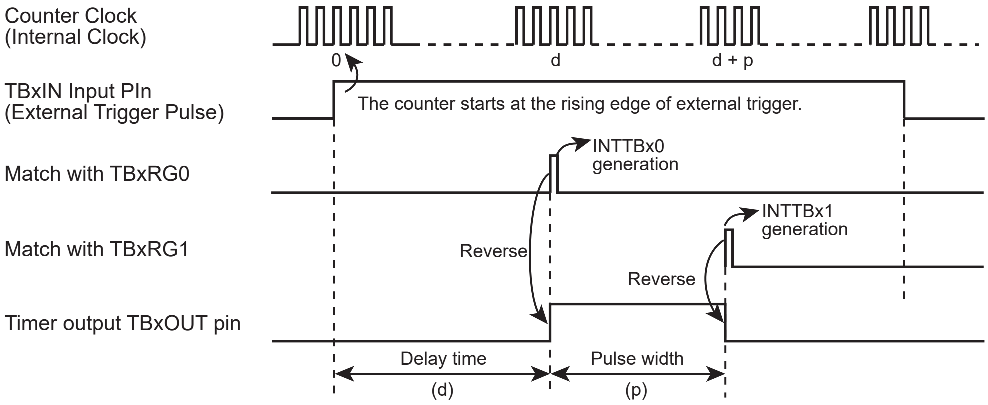


Figure 11-5 One-shot pulse generation using an external count start trigger (with a delay)

11.6 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

11.6.1 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 11-6.

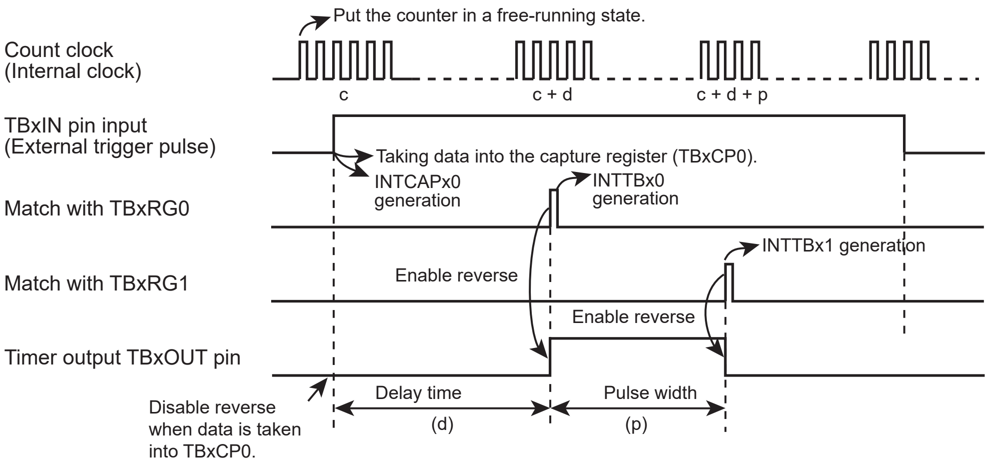


Figure 11-6 One-shot Pulse Output (With Delay)

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

		7	6	5	4	3	2	1	0	
[[Main processing] Capture setting by TBxIN										
Set PORT registers.										Allocates corresponding port to TBxIN.
TBxEN	←	1	X	X	X	X	X	X	X	Enables TMRBx operation.
TBxRUN	←	X	X	X	X	X	0	X	0	Stops count operation.
TBxMOD	←	0	1	0	1	0	0	0	1	Changes source clock to ΦT1. Fetches a count value into the TBxCP0 at the rising edge of TBxIN.
TBxFFCR	←	X	X	0	0	0	0	1	0	Clears TBxFF0 reverse trigger and disables.
Set PORT registers.										Allocates corresponding port to TBxOUT.
Interrupt Set-Enable Register	←	*	*	*	*	*	*	*	*	Permits to generate interrupts specified by INTCAPx0 interrupt corresponding bit by setting to "1".
TBxRUN	←	*	*	*	*	*	1	X	1	Starts the TMRBx module.
[Processing of INTCAPx0 interrupt service routine] Pulse output setting										
TBxRG0	←	*	*	*	*	*	*	*	*	Sets count value.(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 INTTBx1 interrupt corresponding bit setting to "1".
[Processing of INTTBx1 interrupt service routine] Output disable										
TBxFFCR	←	X	X	−	−	0	0	−	−	Clears TBxFF0 reverse trigger setting.
	←	*	*	*	*	*	*	*	*	Prohibits interrupts specified by INTTBx interrupt corresponding bit by setting to "1".

Note: X; Don’t care
−; No change

If a delay is not required, TBxFF0 is reversed when data is taken into TBxCP0, and TBxRG1 is set to the sum of the TBxCP0 value (c) and the one-shot pulse width (p), (c + p), by generating the INTCAPx0 interrupt. (TBxRG1 change must be completed before the next match.)

TBxFF0 is enabled to reverse when UC matches with TBxRG1, and is disabled by generating the INTTBx1 interrupt.

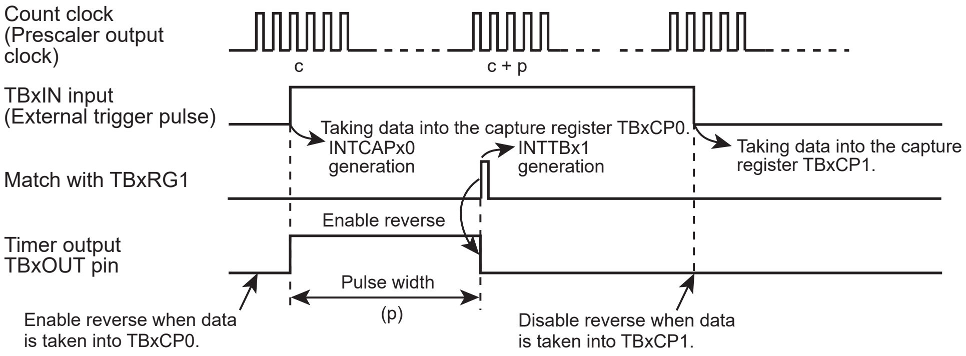

Count clock
(Prescaler output clock)
c
c + p
TBxIN input
(External trigger pulse)
Taking data into the capture register TBxCP0.
INTCAPx0 generation
INTTBx1 generation
Taking data into the capture register TBxCP1.
Match with TBxRG1
Enable reverse
Timer output TBxOUT pin
Pulse width
(p)
Enable reverse when data is taken into TBxCP0.
Disable reverse when data is taken into TBxCP1.

Figure 11-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

11.6.2 Pulse width measurement

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

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

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

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

The "Low" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAPx0 interrupt processing as shown in Figure 11-8 and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

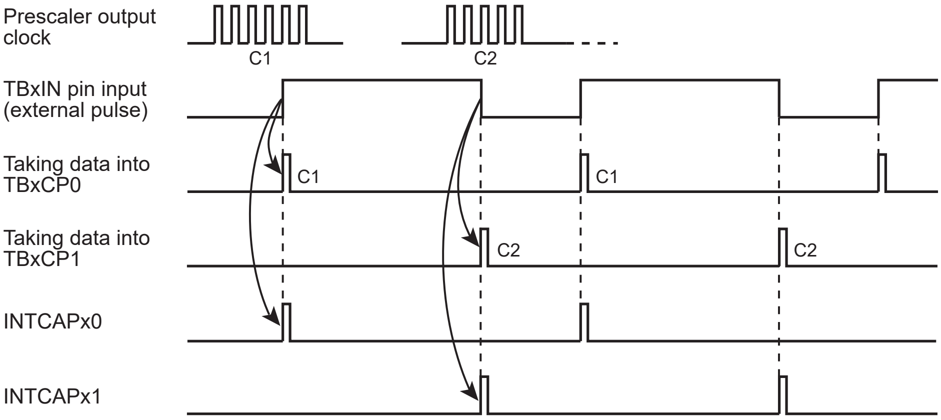


Figure 11-8 Pulse Width Measurement

12. Serial Channel with 4bytes FIFO (SIO/UART)

12.1 Overview

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

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

Their features are given in the following.

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

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

    Keep a "High" level, "Low" level or the state of the last bit
  - The state of SCxTXD pin when an under run error is occurred in clock input mode can be selected as follow:

    Keep a "High" level or "Low" level
  - The last bit hold time of SCxTXD pin can be specified in clock input mode.
- 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 $\overline{\text{SCxCTS}}$ pin
  - Noise cancel for SCxRXD pin

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

12.2 Configuration

Serial channel block diagram and serial clock generator circuit diagram are shown in bellows.

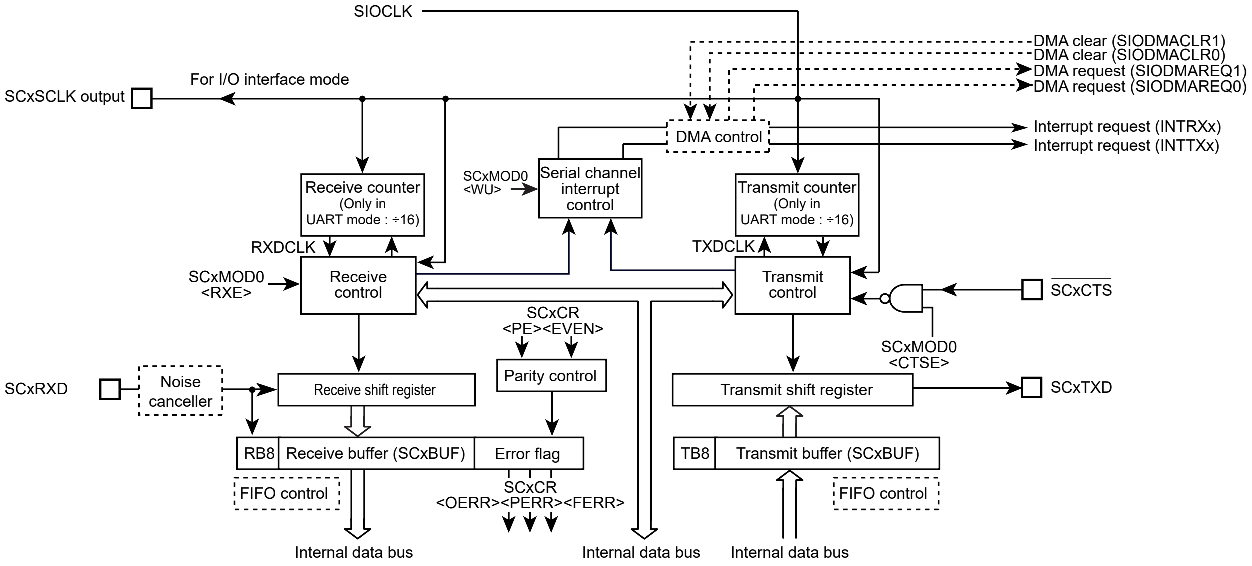


Figure 12-1 Serial Channel Block Diagram

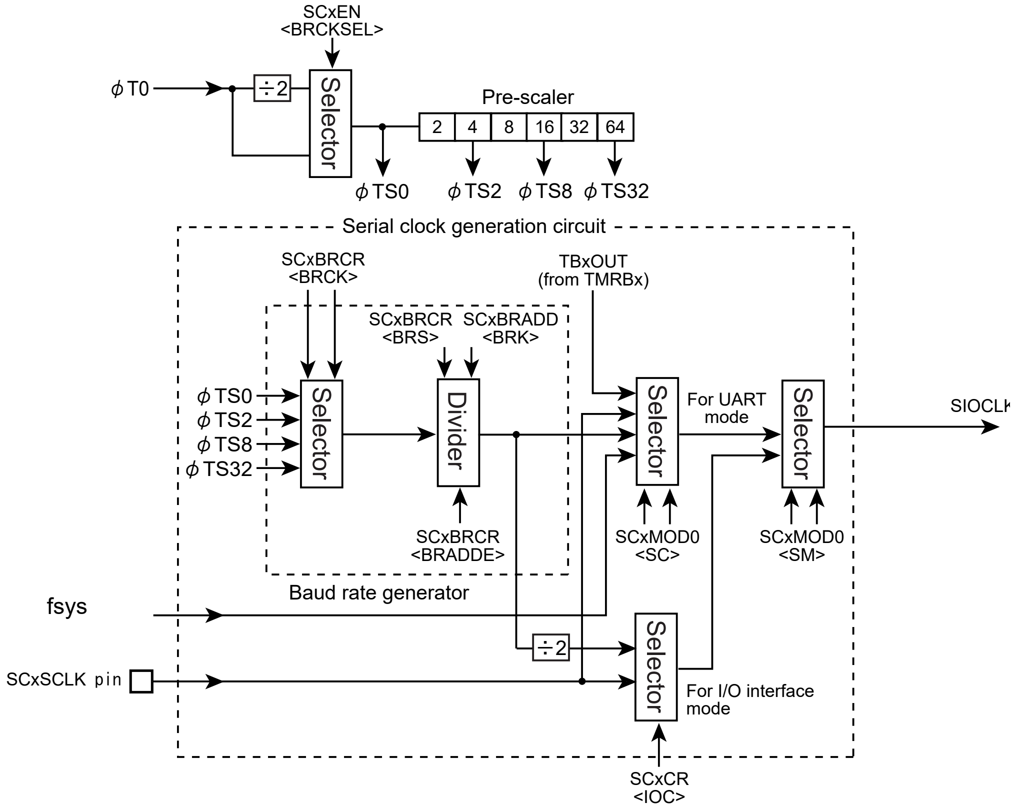


Figure 12-2 Serial clock generation circuit block diagram

12.3 Registers Description

12.3.1 Registers List

The table below shows control registers and their addresses.

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

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

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

12.3.2 SCxEN (Enable Register)

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

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

12.3.3 SCxBUF (Buffer Register)

SCxBUF works as a transmit buffer or FIFO for write operation and as a receive buffer or FIFO for read operation.

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-0	TB[7:0] / RB [7:0]	R/W	[write] TB: Transmit buffer or FIFO [read] RB: Receive buffer or FIFO

12.3.4 SCxCR (Control Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	EHOLD			-	TXDEMP	TIDLE	
After reset	0	0	0	0	0	1	1	0
	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
After reset	0	0	0	0	0	0	0	0

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

Bit	Bit Symbol	Type	Function
1	SCLKS	R/W	Selecting clock edge (For I/O Interface mode) 0: Data in the transmit buffer is sent to SCxTXD pin every one bit on the falling edge of SCxRXD pin. Data from SCxRXD pin is received in the receive buffer every one bit on the rising edge of SCxRXD pin. In this case, the state of a SCxRXD pin starts from "High" level. (Rising edge mode) 1: Data in the transmit buffer is sent to SCxTXD pin every one bit on the rising edge of SCxSCLK pin. Data from SCxRXD pin is received in the receive buffer every one bit on the falling edge of SCxSCLK pin. In this case, the state of a SCxSCLK starts from "Low" level.
0	IOC	R/W	Selecting clock (For I/O Interface mode) 0: Clock output mode (A transfer clock is output from SCxSCLK pin.) 1: Clock input mode (A transfer clock is input to SCxSCLK pin.)

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

12.3.5 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 only UART mode) Writes the 9th bit of transmit data in the 9-bit UART mode.
6	CTSE	R/W	Handshake function control (For only UART mode) 0: CTS disabled 1: CTS enabled Controls handshake function. Setting "1" enables handshake function using $\overline{\text{SCxCTS}}$ pin.
5	RXE	R/W	Receive control (Note1)(Note2) 0: Disabled 1: Enabled
4	WU	R/W	Wake-up function (For only UART mode) 0: Disabled 1: Enabled This function is available only at 9-bit UART mode. In other mode, this function has no meaning. When it is enabled, interrupt is occurred only when RB9 = "1" in a 9-bit UART mode.
3-2	SM[1:0]	R/W	Specifies transfer mode. 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode
1-0	SC[1:0]	R/W	Serial transfer clock (For only UART mode) 00: TMRB output 01: Baud rate generator 10: System clock (fsys) 11: External clock (SCxSCLK pin input) (For the I/O interface mode, the transfer clock in I/O interface mode is selected by SCxCR<IOC>.)

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

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

12.3.6 SCxMOD1 (Mode Control Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	I2SC	FDPX		TXE	SINT			-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	I2SC	R/W	IDLE 0: Stop 1: Operate Specifies operation in the IDLE mode.
6-5	FDPX[1:0]	R/W	Transfer mode setting 00: Transfer prohibited 01: Half duplex (Receive) 10: Half duplex (Transmit) 11: Full duplex Configures the transfer mode in the I/O interface mode. And when FIFO is enabled, specify the configuration of FIFO. In UART mode, specify the only configuration of FIFO.
4	TXE	R/W	Transmit control (Note1)(Note2) 0 :Disabled 1: Enabled This bit enables transmission and is valid for all the transfer modes.
3-1	SINT[2:0]	R/W	Interval time of continuous transmission (For I/O interface mode) 000: None 001: 1 x SCLK cycle 010: 2 x SCLK cycle 011: 4 x SCLK cycle 100: 8 x SCLK cycle 101: 16 x SCLK cycle 110: 32 x SCLK cycle 111: 64 x SCLK cycle This parameter is valid only for the I/O interface mode when SCLK output mode is selected. In other modes, this parameter 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 control registers first and then enable the <TXE>.

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

12.3.7 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". When reading the receive buffer, this bit is cleared to "0".</td></tr>
<tr><td>5</td><td>TXRUN</td><td>R</td><td>In transmission flag
0: Stop
1: Operate
This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> bits indicate the following status.
<table>
<tr><th>&lt;TXRUN&gt;</th><th>&lt;TBEMP&gt;</th><th>Status</th></tr>
<tr><td>1</td><td>−</td><td>Transmission in progress</td></tr>
<tr><td rowspan="2">0</td><td>1</td><td>Transmission is completed.</td></tr>
<tr><td>0</td><td>Wait state with data in transmit buffer</td></tr>
</table></td></tr>
<tr><td>4</td><td>SBLEN</td><td>R/W</td><td>STOP bit length (for UART mode)
0: 1-bit
1: 2-bit
This specifies the length of transmission stop bit in the UART mode.
On the receive side, the decision is made using only a single bit regardless of the <SBLEN&gt;.</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 UART mode, set this bit to LSB first.</td></tr>
<tr><td>2</td><td>WBUF</td><td>R/W</td><td>Enable double-buffer
0: Disabled
1: Enabled
This parameter enables or disables the transmit/receive double buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data i n the I/O interface mode and to transmit in the UART mode.
When receiving data in the I/O interface mode (in clock input mode) and UART mode, double buffering is enabled regardless of the <WBUF&gt;.</td></tr>
</table>

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

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

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

12.3.8 SCxBRCR (Baud Rate Generator Control Register)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	−	R/W	Write "0".
6	BRADDE	R/W	N + (16 − K)/16 divider function (Only for UART mode) 0: disabled 1: enabled
5-4	BRCK[1:0]	R/W	Select input clock to the baud rate generator. 00:φTS0 01:φTS2 10:φTS8 11:φTS32
3-0	BRS[3:0]	R/W	Division ratio "N" 0000: N = 16 0001: N = 1 0010: N = 2 ... 1111: N = 15

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

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

12.3.9 SCxBRADD (Baud Rate Generator Control Register 2)

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

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

Table 12-1 lists the settings of baud rate generator division ratio.

Table 12-1 Setting division ratio

	<BRADDE> = "0"	<BRADDE> = "1" (Note1) (Only in the UART mode)
<BRS>	Specify "N"	
<BRK>	No setting required	Specify "K" (Note2)
Division ratio	Divide by N	$N + \frac{(16-K)}{16}$ division.

Note 1: To use the "N + (16 - K)/16" division function, be sure to set <BRADDE> to "1" after setting the K value to <BRK>. The "N + (16 - K)/16" division function can only be used in the UART mode.

Note 2: Specifying "K = 0" is prohibited.

12.3.10 SCxFCNF (FIFO Configuration Register)

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

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

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

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

12.3.11 SCxRFC (Receive FIFO Configuration Register)

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

<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>RFCS</td><td>W</td><td>Receive FIFO clear (Note)
1: Clear
When SCxRFC<RFCS> is set to "1", the receive FIFO is cleared and SCxRST<RLVL[2:0]> is "000". And also the read pointer is initialized.
Read as "0".</td></tr>
<tr><td>6</td><td>RFIS</td><td>R/W</td><td>Select interrupt generation condition.
0: When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt (<RIL [1:0]>)
1: When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt (<RIL [1:0]>)
For the detail of interrupt condition, refer to "12.13.1.2 FIFO"</td></tr>
<tr><td>5-2</td><td>-</td><td>R</td><td>Read as "0".</td></tr>
<tr><td>1-0</td><td>RIL[1:0]</td><td>R/W</td><td>FIFO fill level to generate receive interrupts.
.
<table>
<tr><th></th><th>Half duplex</th><th>Full duplex</th></tr>
<tr><td>00</td><td>4 bytes</td><td>2 bytes</td></tr>
<tr><td>01</td><td>1 byte</td><td>1 byte</td></tr>
<tr><td>10</td><td>2 bytes</td><td>2 bytes</td></tr>
<tr><td>11</td><td>3 bytes</td><td>1 byte</td></tr>
</table></td></tr>
</table>

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

12.3.12 SCxTFC (Transmit FIFO Configuration Register)

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

Bit	Bit Symbol	Type	Function
31-9	−	R	Read as "0".
8	TBCLR	W	Transmit buffer clear 0: Don't care 1: Clear When SCxTFC<TBCLR> is set to "1", the transmit buffer is cleared. Read as "0".
7	TFCS	W	Transmit FIFO clear (Note1) 0: Don't care 1: Clear When SCxTFC<TFCS> is set to "1", the transmit FIFO is cleared and SCxTST<TLVL[2:0]> is "000". And also the write pointer is initialized. Read as "0".
6	TFIS	R/W	Selects interrupt generation condition. 0: When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt (<TIL [1:0]>) 1: When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt (<TIL [1:0]>) For the detail of interrupt condition, refer to "12.13.2.2 FIFO"
5-2	−	R	Read as "0".
1-0	TIL[1:0]	R/W	Fill level which transmit interrupt is occurred.

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

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

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

12.3.13 SCxRST (Receive FIFO Status Register)

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

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

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

12.3.14 SCxTST (Transmit FIFO Status Register)

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

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

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

12.4 Operation in Each Mode

Table 12-2 shows the modes.

Table 12-2 Modes

Mode	type	Data length	Transfer direction	Specifies whether to use parity bits.	STOP bit length (transmit)
Mode 0	Synchronous communication mode (I/O interface mode)	8 bits	LSB first/MSB first	-	-
Mode 1	Asynchronous communication mode (UART mode)	7 bits	LSB first	o	1 bit or 2 bits
Mode 2		8 bits		o	
Mode 3		9 bits		×	

The Mode 0 is a synchronous communication and can be used to extend I/O. This mode transmits and receives data in synchronization with SCLK clock. SCLK clock can be used for both input and output modes. The direction of data transfer can be selected from LSB first or 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 directions can be selected as only 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.

12.5 Data Format

12.5.1 Data Format List

Figure 12-3 shows data format.

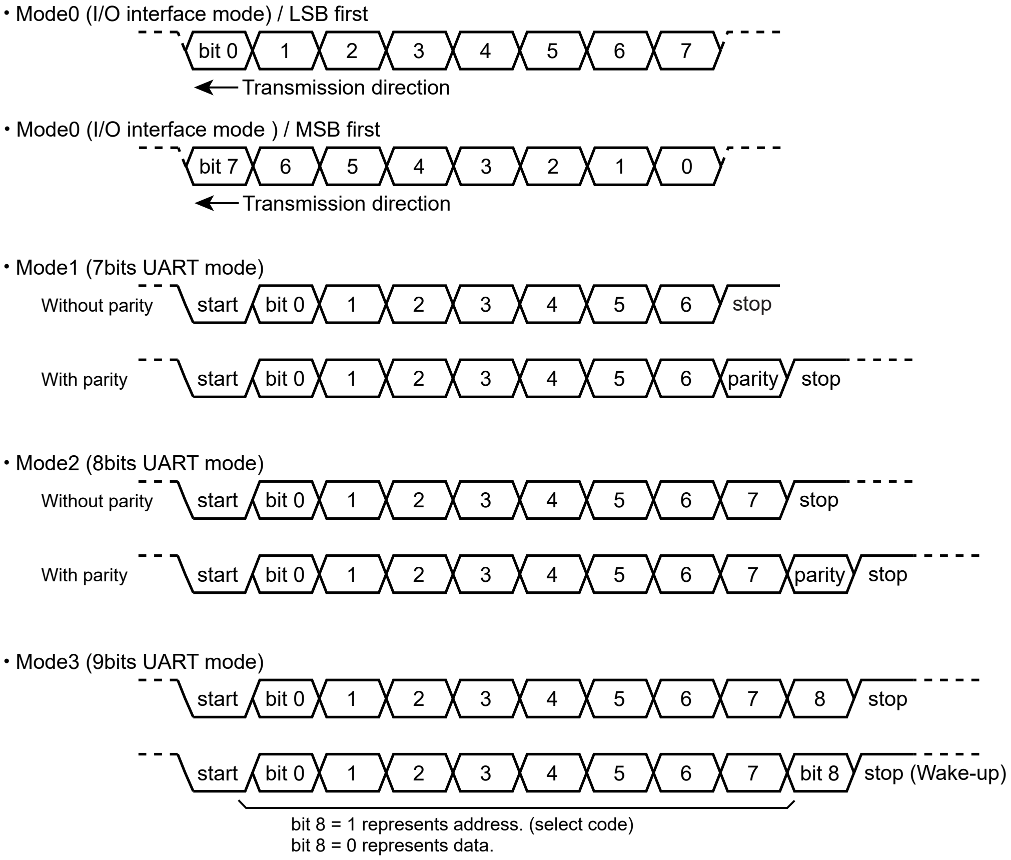


Figure 12-3 Data Format

12.5.2 Parity Control

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

Setting "1" to SCxCR<PE> enables the parity. SCxCR<EVEN> selects either even or odd parity.

12.5.2.1 Transmission

Upon data transmission, the parity control circuit automatically generates the parity with the data in the transmit buffer. The parity bit will be stored in SCxBUF<TB7> in the 7-bit UART mode and SCxMOD<TB8> in the 8-bit UART mode.

The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

12.5.2.2 Reception

If the received data is moved from the receive shift register to the receive buffer, a parity is generated.

In the 7-bit UART mode, the generated parity is compared with the parity stored in SCxBUF<RB7>, in the 8-bit UART mode, it is compared with the one in SCxCR<RB8>.

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

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

12.5.3 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.

12.6 Clock Control

12.6.1 Prescaler

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

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

12.6.2 Serial Clock Generation Circuit

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

12.6.2.1 Baud Rate Generator

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

(1) Baud Rate Generator input clock

The input clock of the baud rate generator is selected from the prescaler outputs divided by 1, 4, 16 and 64.

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

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

(2) Baud Rate Generator output clock

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

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

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

Mode	Divide Function Setting SCxBRCR<BRADDE>	Divide by N SCxBRCR<BRS[3:0]>	Divide by K SCxBRADD<BRK[3:0]>
I/O interface	Divide by N	1 to 16 (Note)	-
UART	Divide by N	1 to 16	-
	N + (16-K)/16 division	2 to 15	1 to 15

Note: 1/N (N=1) frequency division ratio can be used only when a double buffer is enabled.

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

- Divide by N

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

- N + (16-K)/16 division

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

12.6.2.2 Clock Selection Circuit

A clock can be selected by setting the modes and the register.

Modes can be specified by setting the SCxMOD0<SM[1:0]>

The clock in I/O interface mode is selected by setting SCxCR<IOC><SCLKS>.

The clock in UART mode is selected by setting SCxMOD0<SC[1:0]>.

(1) Transfer Clock in I/O interface mode

Table 12-3 shows clock selection in I/O interface mode.

Table 12-3 Clock Selection in I/O Interface Mode

Mode SCxMOD0<SM[1:0]>	Input/Output selection SCxCR<IOC>	Clock edge selection SCxCR<SCLKS>	Clock of use
"00" (I/O interface mode)	"0" (Clock output mode)	"0" (Transmit : falling edge, Receive : rising edge)	Divided by 2 of the baud rate generator output.
		"1" (Transmit : rising edge, Receive : falling edge)	Divided by 2 of the baud rate generator output.
	"1" (Clock input mode)	"0" (Transmit : falling edge, Receive : rising edge)	SCxSCLK pin input
		"1" (Transmit : rising edge, Receive : falling edge)	SCxSCLK pin input

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

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

(2) Transfer clock in the UART mode

Table 12-4 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 12-4 Clock Selection in UART Mode

Mode SCxMOD0<SM[1:0]>	Clock selection SCxMOD0<SC[1:0]>
UART Mode ("01", "10", "11")	"00" : TMRB output
	"01" : Baud rate generator
	"10" : fsys
	"11" : SCxSCLK pin input

To use SCxSCLK pin input, the following conditions must be satisfied.

- SCLK cycle > 2/fsys

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

Baud rates can be obtained by using the following formula.

Baud rate calculation

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[2:0]>}}{\underbrace{(\text{TBxRG1} \times 2)}_{\text{One clock cycle is a period that the timer flip-flop is inverted twice.}} \times \underbrace{2}_{\text{In the case the timer prescaler clock } \Phi \text{ T1 (2 division ratio) is selected.}} \times 16}$$

12.7 Transmit/Receive Buffer and FIFO

12.7.1 Configuration

Figure 12-4 shows the configuration of transmit buffer, receive buffer and FIFO.

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

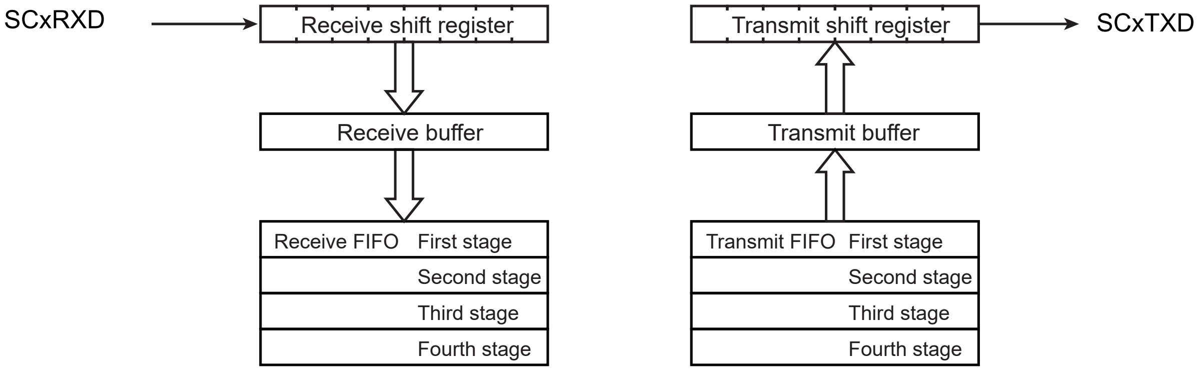


Figure 12-4 The Configuration of Buffer and FIFO

12.7.2 Transmit/Receive Buffer

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

When serial channel is operated as receive, if it is operated as clock input mode in the I/O interface mode or it is operated as the UART mode, it’s double buffered regardless of <WBUF> settings.

In other modes, it’s according to the <WBUF> settings.

Table 12-5 shows correlation between modes and buffers.

Table 12-5 Mode and buffer Composition

Mode		SCxMOD2<WBUF>	
		"0"	"1"
UART mode	Transmit	Single	Double
	Receive	Double	Double
I/O interface mode (Clock input mode)	Transmit	Single	Double
	Receive	Double	Double
I/O interface mode (Clock output mode)	Transmit	Single	Double
	Receive	Single	Double

12.7.3 Initialize Transmit Buffer

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

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

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

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

12.7.4 FIFO

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

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

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

Table 12-6 shows correction between modes and FIFO.

Table 12-6 Mode and FIFO Composition

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

12.8 Status Flag

The SCxMOD2 has two types of flag. This bit is significant only when the double buffer is enabled.

<RBFLL> is a flag to show that the receive buffer is full. When one frame of data is received and the data is moved from the receive shift register to the receive buffers, this bit changes to "1". When reading the receive buffer is read, this bit is cleared to "0".

<TBEMP> shows that the transmit buffer is empty. When data in the transmit buffers is moved to the transmit shift register, this bit is set to "1". When data is set to the transmit buffers, the bit is cleared to "0".

12.9 Error Flag

Three error flags are provided in the SCxCR. 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.

Mode	Flag		
	<OERR>	<PERR>	<FERR>
UART mode	Over-run error	Parity error	Framing error
I/O Interface mode (Clock input mode)	Over-run error	Under-run error (When a double buffer and FIFO are used)	Fixed to 0
		Fixed to 0 (When a double buffer and FIFO are not used)	
I/O Interface mode (Clock output mode)	Undefined	Undefined	Fixed to 0

12.9.1 OERR Flag

In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame before the receive buffer has been read.

If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied).

In the I/O interface mode with clock output mode, the SCxSCLK pin output stops upon setting the flag.

Note: **To switch from the I/O interface mode with clock output mode to other modes, read the SCxCR and clear the overrun flag.**

12.9.2 PERR Flag

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

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

In the I/O interface mode, <PERR> is set to "1" under the following conditions when a double buffer is enabled.

In the clock input mode, <PERR> is set to "1" when the clock is input after completing data output of the transmit shift register with no data in the transmit buffer.

In the clock output mode, <PERR> is set to "1" after completing output of all data and the clock output stops.

Note: To switch from the I/O interface mode with clock output mode to other modes, read the SCxCR and clear the under-run flag.

12.9.3 FERR Flag

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

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

12.10 Receive

12.10.1 Receive Counter

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

In the UART mode, sixteen SIOCLK clock pulses are used in receiving a single data bit and the data symbol is sampled at the eighth pulse.

12.10.2 Receive Control Unit

12.10.2.1 I/O interface mode

In the clock output mode with SCxCR <IOC> set to "0", the SCxRXD pin is sampled on the rising or falling edge of SCxSCLK pin depending on the SCxCR <SCLKS>.

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

12.10.2.2 UART Mode

The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

12.10.3 Receive Operation

12.10.3.1 Receive Buffer

The received data is stored by 1 bit in the receive shift register. When a complete set of bits has been stored, the interrupt INTRXx is generated.

When the double buffer is enabled, the data is moved to the receive buffer (SCxBUF) and the receive buffer full flag (SCxMOD2<RBFLL>) is set to "1". The receive buffer full flag is cleared to "0" by reading the receive buffer. When the double buffer is disabled, the receive buffer full flag has no meaning.

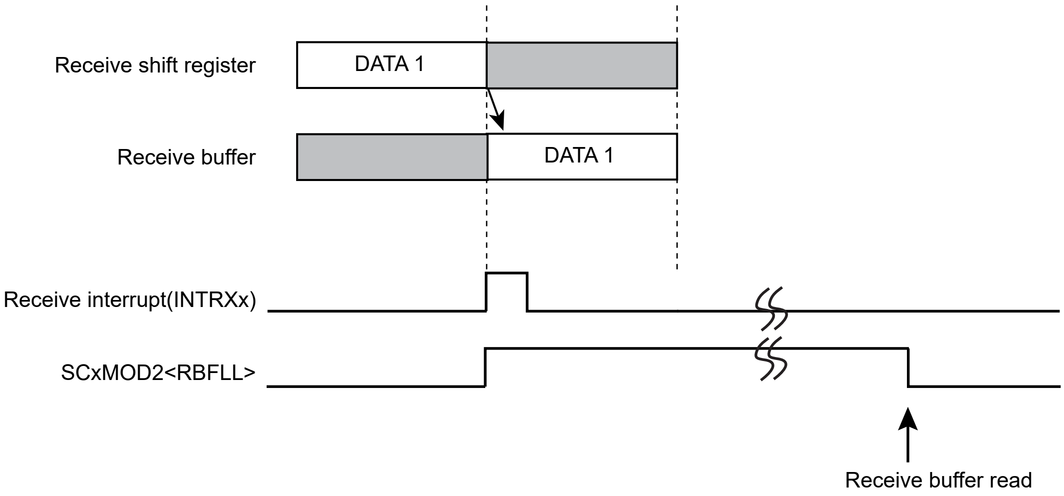


Figure 12-5 Receive Buffer Operation

12.10.3.2 Receive FIFO Operation

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

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

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

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

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

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

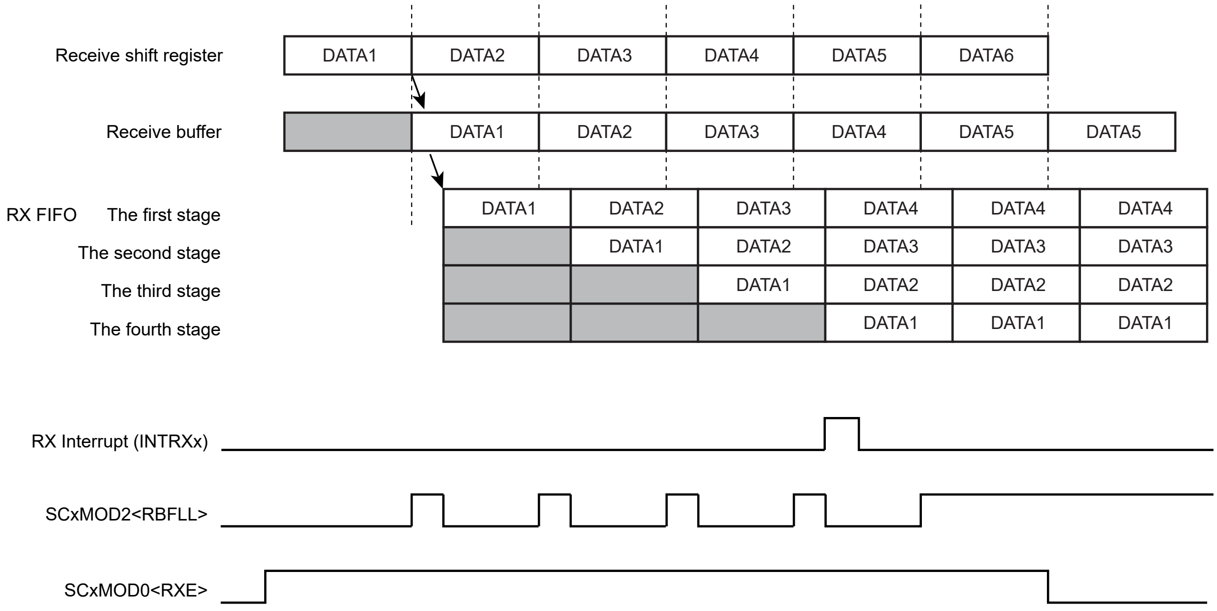


Figure 12-6 Receive FIFO Operation

12.10.3.3 I/O interface mode with clock output mode

In the I/O interface mode with clock output mode setting, clock stops when all received data is stored in the receive buffer and FIFO. So, in this mode, the over-run error flag has no meaning.

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

(1) Case of single buffer

Stop clock 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, clock output is restarted.

(2) Case of double buffer

Stop clock output after receiving the data into a receive shift register and a receive buffer.

When a data is read, clock output is restarted.

(3) Case of FIFO

Stop clock output after receiving the data into a shift register, received buffer and FIFO.

When one byte data is read, the data in the received buffer is transferred into FIFO and the data in the receive shift register is transferred into the received buffer and clock output restarts.

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

12.10.3.4 Read Received Data

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

When receive FIFO is disabled, the buffer full flag SCxMOD2<RBFLL> is cleared to "0" by this reading. The next data can be received in the receive shift register before reading a data from the receive buffer. The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SCxCR<RB8>.

When the receive FIFO is enabled, the 9-bit UART mode is prohibited because up to 8-bit data can be stored in receive FIFO. In the 8-bit UART mode, the parity bit is lost but parity error is determined and the result is stored in SCxCR<PERR>.

12.10.3.5 Wake-up Function

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SCxMOD0 <WU> to "1". In this case, the interrupt INTRXx will be generated only when SCxCR <RB8> is set to "1".

12.10.3.6 Overrun Error

When receive FIFO is disabled, the overrun error occurs without completing reading data before receiving the next data. When an overrun error occurs, a content of receive buffer and SCxCR<RB8> is not lost, but a content of receive shift register is lost.

When receive FIFO is enabled, overrun error is occurred and set overrun flag by no reading receive FIFO before moving the next data into received buffer when receive FIFO is full. In this case, the contents of receive FIFO are not lost.

In the I/O interface mode with clock output mode, the clock output automatically stops, so this flag has no meaning.

Note: When the mode is changed from I/O interface mode with clock output mode to the other modes, read SCxCR and clear overrun flag.

12.11 Transmit

12.11.1 Transmit Counter

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

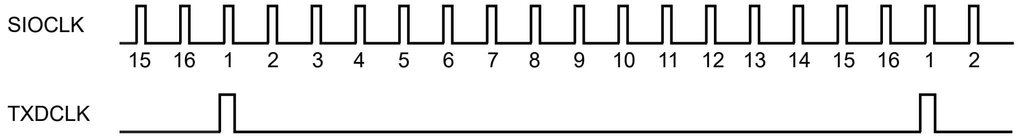


Figure 12-7 Generation of Transmission Clock in UART mode

12.11.2 Transmit Control

12.11.2.1 In I/O Interface Mode

In the clock output mode with SCxCR<IOC> set to "0", each bit of data in the transmit buffer is outputted to the SCxTXD pin on the rising or falling edge of SCxSCLK pin according to the SCxCR<SCLKS>.

In the clock input mode with SCxCR<IOC> set to "1", each bit of data in the transmit buffer is outputted to the SCxTXD pin on the rising or falling edge of the SCxSCLK pin according to the SCxCR<SCLKS>.

12.11.2.2 In 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.

12.11.3 Transmit Operation

12.11.3.1 Operation of Transmit 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.

When double buffering is enabled (including the case the transmit FIFO is enabled), if "1" is set to SCxMOD1<TXE>, data in the transmit buffer is transferred 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".

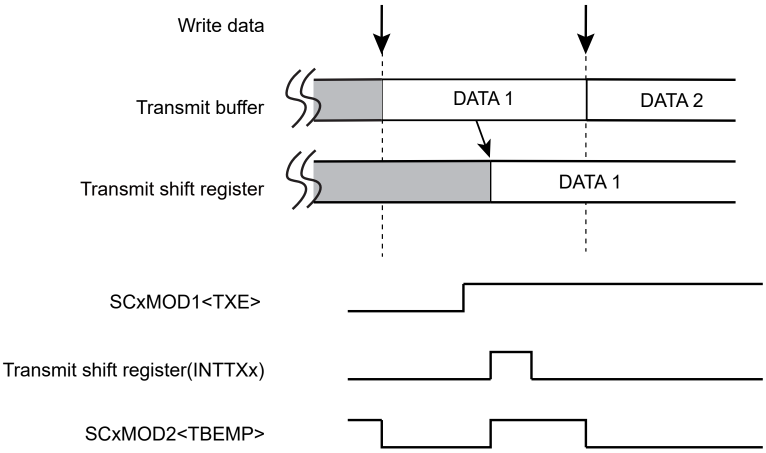


Figure 12-8 Operation of Transmit Buffer (Double-buffer is enabled)

12.11.3.2 Transmit FIFO Operation

When FIFO is enabled, the maximum 5-byte data can be stored using the transmit buffer and FIFO. Once transmission is enabled, data is transferred to the transmit shift register from the transmit buffer and start transmission. If data exists in the FIFO, the data is moved to the transmit buffer immediately, and the <TBEMP> flag is cleared to "0".

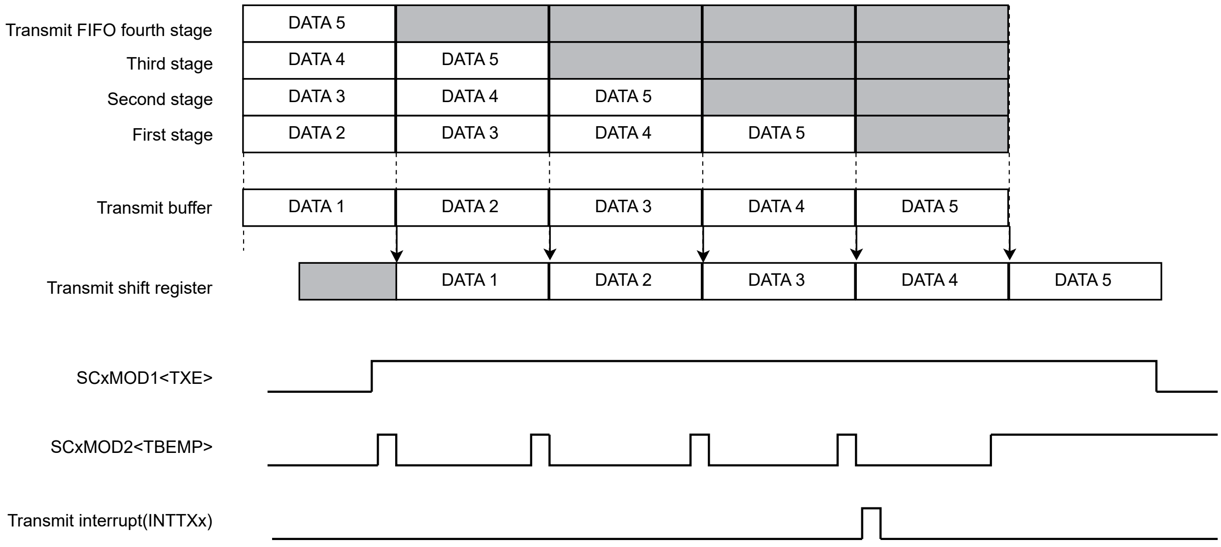
Note: To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").

Settings and operations to transmit 5 bytes data stream by setting the transfer mode to half duplex are shown as below.

SCxMOD1<FDPX[1:0]> = "10"	:Transfer mode is set to half duplex.
SCxFCNF<RFST><TFIE><RFIE>	:Transmission is automatically disabled if FIFO becomes empty.
<RXTXCNT><CNFG> = "11011"	:The number of bytes to be used in the receive FIFO is the same as the interrupt
	:generation fill level.
SCxTFC<TIL[1:0]> = "00"	:Sets the interrupt generation fill level to "0".
SCxTFC<TFCS><TFIS> = "11"	:Clears receive FIFO and sets the condition of interrupt generation.
SCxFCNF<CNFG> = "1"	:Enable FIFO

After above settings are configured, data transmission can be initiated by writing 5 bytes of data to the transmit buffer and FIFO, and setting the SCxMOD1<TXE> bit to "1". When the last transmit data is moved to the transmit buffer, the transmit interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.

Once above settings are configured, if the transmission is not set as auto disabled, the transmission should last writing transmit data.



12.11.3.3 Transmit in I/O interface Mode with Clock Output Mode

In the I/O interface mode with clock output mode, the clock output automatically stops when all data transmission is completed and underrun error will not occur.

The timing of suspension and resume of clock output is different depending on the buffer and FIFO usage.

(1) Single Buffer

The clock output stops each time one frame of data is transferred. Handshaking for each data with the other side of communication can be enabled. The clock output resumes when the next data is written in the buffer.

(2) Double Buffer

The clock output stops upon completion of data transmission in the transmit shift register and the transmit buffer. The clock 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, clock output resumes.

If SCxFCNF<RXTXCNT> is configured, SCxMOD0<TXE> bit is cleared at the same time as clock stops and the transmission stops.

12.11.3.4 Level of SCxTXD pin after the last bit is output in I/O interface mode

The level of SCxTXD pin after the data hold time is passed after the last bit is output is specified by SCxCR<TIDLE>.

When SCxCR<TIDLE> is "00", the level of SCxTXD pin is output "Low" level. When SCxCR<TIDLE> is "01", the level of SCxTXD pin is output "High" level. When SCxCR<TIDLE> is "10", the level of SCxTXD pin is output the level of the last bit.

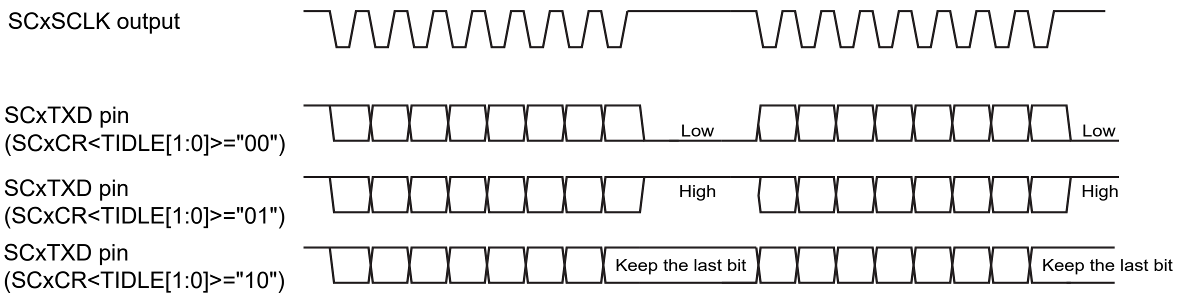


Figure 12-9 Level of SCxTXD pin After the last bit is output

12.11.3.5 Under-run error

In the I/O interface mode with clock input mode and if FIFO is empty and if no data is set in transmit buffer before the next frame clock input, which occurs upon completion of data transmission from transmit shift register, an under-run error occurs and SCxCR<PERR> is set to "1".

The level of a SCxTXD pin can be specified by SCxCR<TXDEMP>. When SCxCR<TXDEMP> is "0", a SCxTXD pin outputs "Low" level during data output period. When SCxCR<TXDEMP> is "1", a SCxTXD pin outputs "High" level.

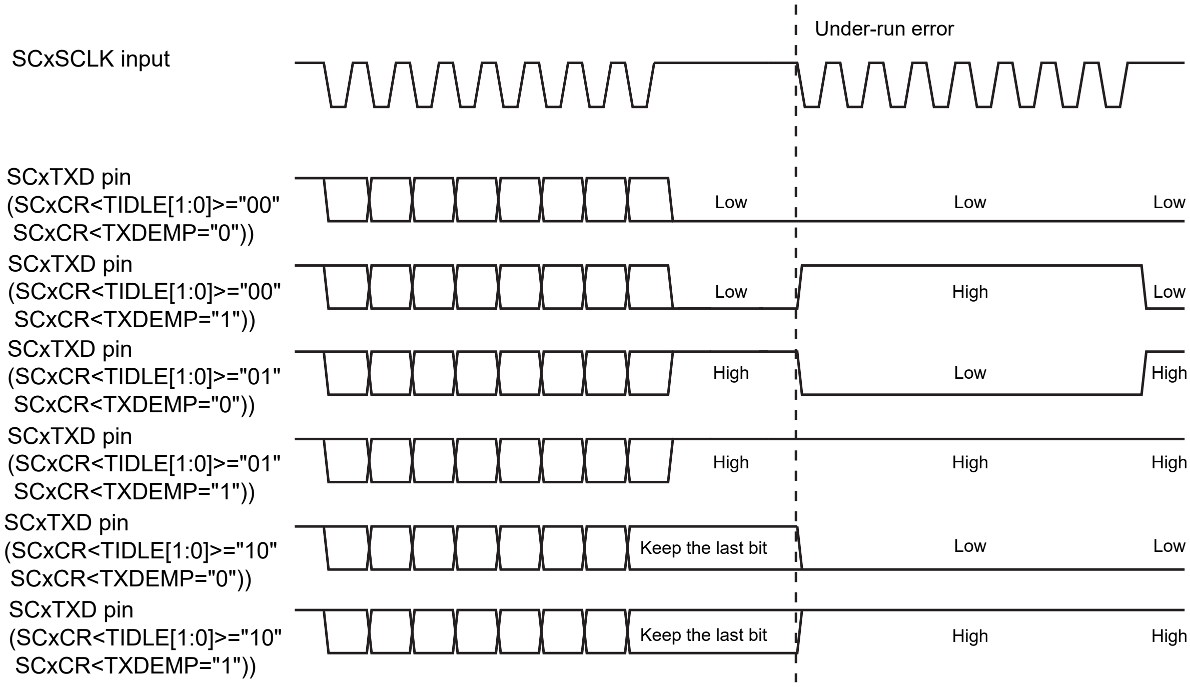

Under-run error
SCxSCLK input
SCxTXD pin (SCxCR<TIDLE[1:0]>="00" SCxCR<TXDEMP="0"))
Low
Low
Low
SCxTXD pin (SCxCR<TIDLE[1:0]>="00" SCxCR<TXDEMP="1"))
Low
High
Low
SCxTXD pin (SCxCR<TIDLE[1:0]>="01" SCxCR<TXDEMP="0"))
High
Low
High
SCxTXD pin (SCxCR<TIDLE[1:0]>="01" SCxCR<TXDEMP="1"))
High
High
High
SCxTXD pin (SCxCR<TIDLE[1:0]>="10" SCxCR<TXDEMP="0"))
Keep the last bit
Low
Low
SCxTXD pin (SCxCR<TIDLE[1:0]>="10" SCxCR<TXDEMP="1"))
Keep the last bit
High
High

Figure 12-10 Level of SCxTXD pin when Under-run Error is Occurred

In the I/O interface mode with SCLK output setting, the clock output automatically stops, so SCxCR<PERR> has no meaning.

Note: Before switching the I/O interface mode with clock output mode to other modes, read the SCxCR and clear the under-run flag.

12.11.3.6 Data Hold Time In the I/O interface mode with clock input mode

In the I/O interface mode with clock input mode, a data hold time of the last bit can be adjusted by SCxCR<EHOLD[2:0]>. Specify a data hold time and the period of the SCLK to satisfy the following formula.

The data hold time of the last bit ≤ The period of SCLK / 2

12.12 Handshake function

The function of the handshake is to enable frame-by-frame data transmission by using the $\overline{\text{SCxCTS}}$ (Clear to send) pin and to prevent over-run errors. This function can be enabled or disabled by SCxMOD0<CTSE>.

When the $\overline{\text{SCxCTS}}$ pin is set to "High" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{SCxCTS}}$ pin returns to the "Low" level. The INTTXx interrupt is generated in the normal timing, the next transmit data is written in the transmit buffer, and it waits until it is ready to transmit data.

Note 1: If the $\overline{\text{CTS}}$ signal is set to "High" level during transmission, the next data transmission is suspended after the current transmission is completed.

Note 2: Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{\text{CTS}}$ is set to "Low" level.

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.

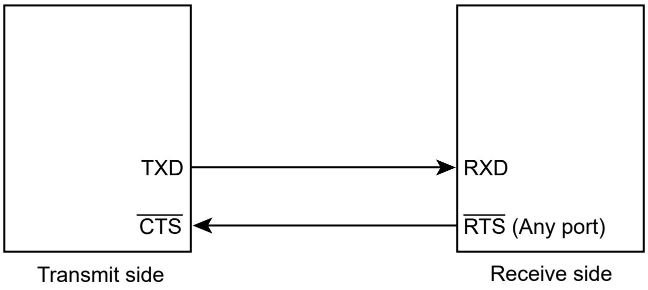


Figure 12-11 Handshake Function

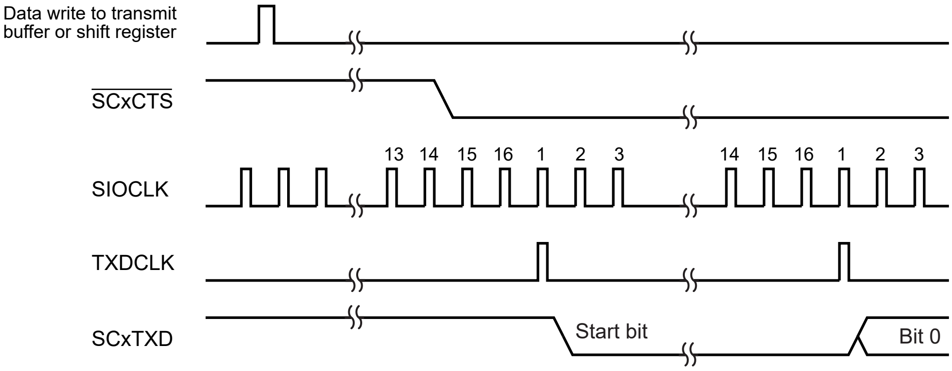


Figure 12-12 $\overline{\text{SCxCTS}}$ Signal timing

12.13 Interrupt/Error Generation Timing

12.13.1 Receive Interrupts

Figure 12-13 shows the data flow of receive operation and the route of read.

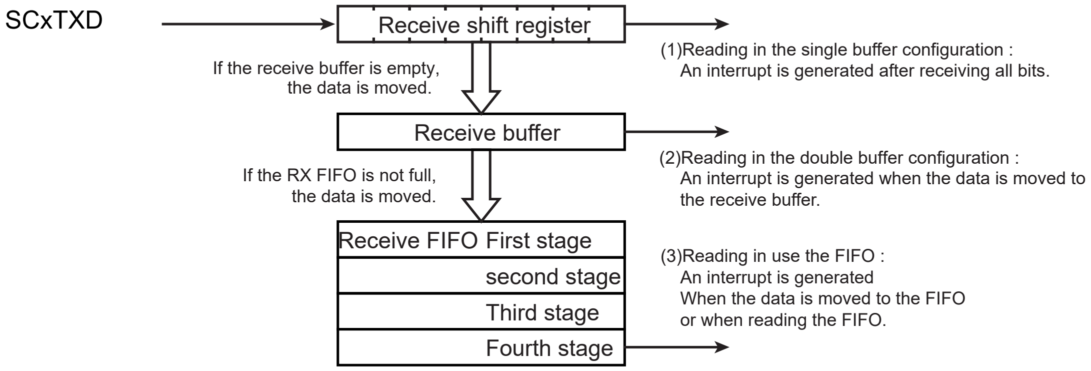

SCxTXD
Receive shift register
If the receive buffer is empty, the data is moved.
Receive buffer
If the RX FIFO is not full, the data is moved.
Receive FIFO First stage
second stage
Third stage
Fourth stage
(1)Reading in the single buffer configuration : An interrupt is generated after receiving all bits.
(2)Reading in the double buffer configuration : An interrupt is generated when the data is moved to the receive buffer.
(3)Reading in use the FIFO : An interrupt is generated When the data is moved to the FIFO or when reading the FIFO.

Figure 12-13 Receive Buffer/FIFO Configuration Diagram

12.13.1.1 Single Buffer / Double Buffer

Receive interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Table 12-7 Receive Interrupt Conditions in use of Single Buffer / Double Buffer

Buffer Configurations	UART modes	IO interface modes
Single Buffer	−	Immediately after the raising / falling edge of the last SCxSCLK pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	A receive interrupt occurs when data is transferred from the receive shift register to the receive buffer. Specific timings are : ・If data does not exist in the receive buffer, a receive interrupt occurs in the vicinity of the center of the 1st stop bit. ・If data exists in both the receive shift register and the receive buffer, a receive interrupt occurs when the buffer is read.	A receive interrupt occurs when data is transferred from the receive shift register to the receive buffer. Specific timings are: ・If data does not exit int the receive buffer, a receive interrupt occurs immediately after on rising/falling edge of SCxSCLK pin of the last bit. (The setting of rising edge or falling edge is specified with SCxCR<SCLKS>.) ・If data exists in both the receive shift register and the receive buffer, a receive interrupt occurs when the buffer is read.

Note: Interrupts are not generated when an over-run error is occurred.

12.13.1.2 FIFO

When the FIFO is used, a receive interrupt occurs on depending on the timing described in Table 12-8 and the condition specified with SCxRFC<RFIS>.

Table 12-8 Receive Interrupt Conditions in use of FIFO

SCxRFC<RFIS>	Interrupt conditions	Interrupt generation timing
"0"	When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	・When transfer a received data from receive buffer to receive FIFO
"1"	When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt <RIL[1:0]>	・When read a receive data from receive FIFO ・When transfer a received data from receive buffer to receive FIFO

12.13.2 Transmit interrupts

Figure 12-14 shows the data flow of transmit operation and the route of read.

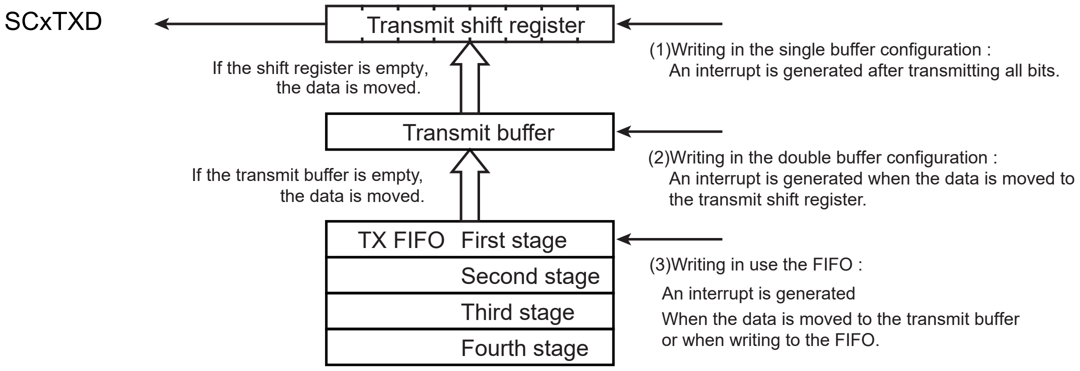

SCxTXD
Transmit shift register
If the shift register is empty, the data is moved.
(1)Writing in the single buffer configuration :
An interrupt is generated after transmitting all bits.
Transmit buffer
If the transmit buffer is empty, the data is moved.
(2)Writing in the double buffer configuration :
An interrupt is generated when the data is moved to the transmit shift register.
TX FIFO First stage
Second stage
Third stage
Fourth stage
(3)Writing in use the FIFO :
An interrupt is generated
When the data is moved to the transmit buffer or when writing to the FIFO.

Figure 12-14 Transmit Buffer / FIFO Configuration Diagram

12.13.2.1 Singe Buffer / Double Buffer

Transmit interrupts are generated at the time depends on the transfer mode and the buffer configurations, which are given as follows.

Table 12-9 Transmit Interrupt conditions in use of Single Buffer/Double Buffer

Buffer Configurations	UART modes	IO interface modes
Single Buffer	Just before the stop bit is sent	Immediately after the raising / falling edge of the last SCxSCLK pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Double Buffer	When a data is moved from the transmit buffet to the transmit shift register. If "1" is set to SCxMOD1<TXE> and the transmit shift register is empty, a transmit interrupt occurs because data is immediately transferred to the transmit shift register from the transmit buffer.	

12.13.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 12-10 and the condition specified with SCxTFC<TFIS>.

Table 12-10 Transmit Interrupt conditions in use of FIFO

SCxTFC<TFIS>	Interrupt condition	Interrupt generation timing
"0"	When FIFO fill level (SCxTST<TLVL[2:0]>) = Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>	・When transmitted data is transferred from transmit FIFO to transmit buffer
"1"	When FIFO fill level (SCxTST<TLVL[2:0]>) ≤ Transmit FIFO fill level to generate transmit interrupt <TIL[1:0]>	・When transmit data is write into transmit FIFO ・When transmitted data is transferred from transmit FIFO to transmit buffer

12.13.3 Error Generation

12.13.3.1 UART Mode

Error	9 bits	7 bits 8 bits 7 bits + Parity 8 bits + Parity
Framing Error over-run Error	Around the center of stop bit	
Parity Error	−	Determination : Around the center of parity bit Flag-change : Around the center of stop bit

12.13.3.2 I/O Interface Mode

over-run Error	Immediately after the raising / falling edge of the last SCxSCLK pin (Rising or falling is determined according to SCxCR<SCLKS> setting.)
Under-run Error	Immediately after the rising or falling edge of the next SCxSCLK pin. (Rising or falling is determined according to SCxCR<SCLKS> setting.)

Note: Over-run error and Under-run error have no meaning in clock output mode.

12.15 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 and the transmit circuit become initial state.

Other states are maintained.

12.16 Operation in Each Mode

12.16.1 Mode 0 (I/O interface mode)

The I/O interface mode is selected by setting SCxMOD<SM[1:0]> to "00".

Mode 0 consists of two modes, the clock output mode to output synchronous clock (SCLK) and the clock input mode to accept synchronous clock (SCLK) from an external source.

The operation with disabling a FIFO in each mode is described below. Regarding a FIFO, refer to a receive FIFO and a transmit FIFO which are described before.

12.16.1.1 Transmit

(1) Clock Output Mode

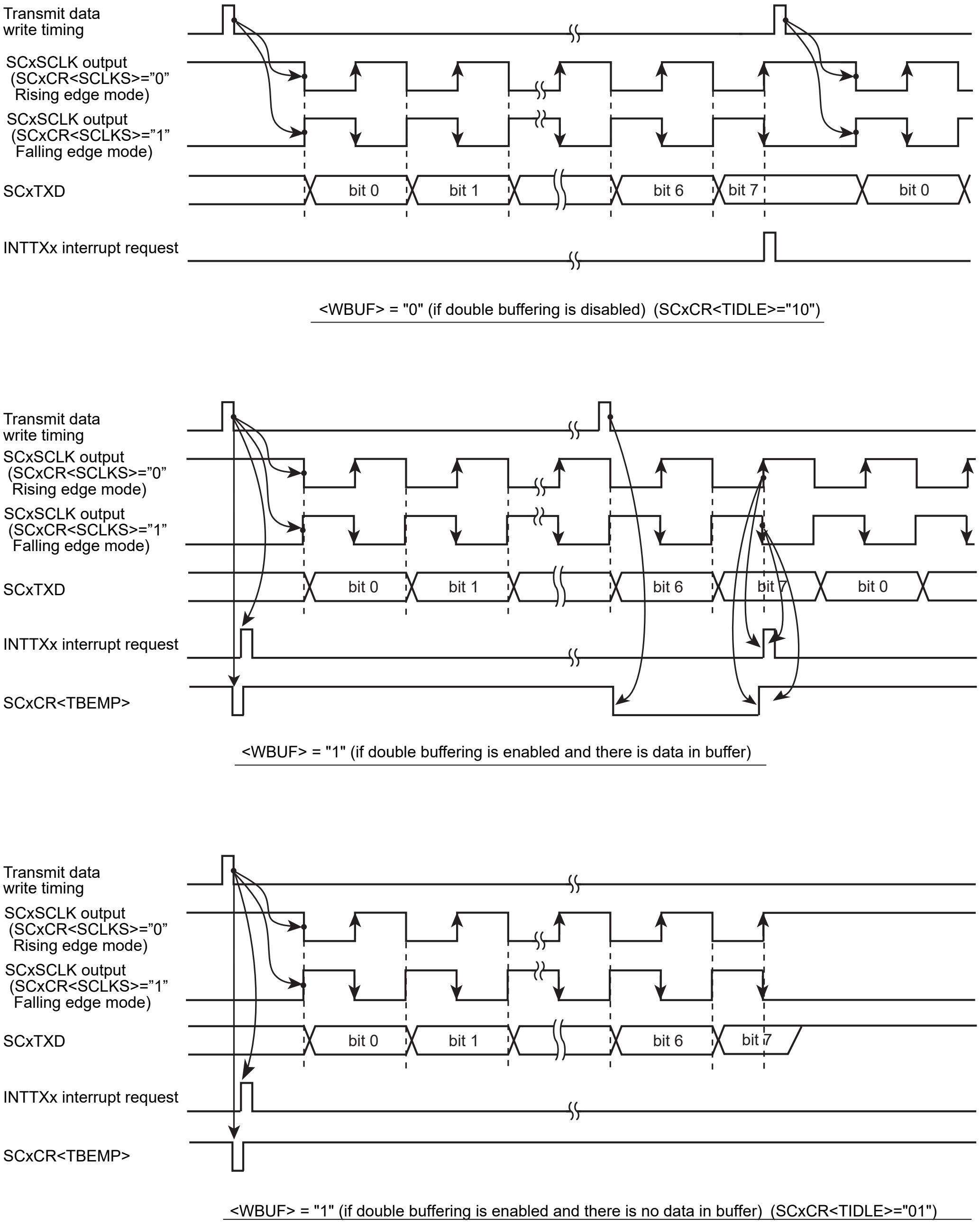
- If the transmit double buffer is disabled (SCxMOD2<WBUF> = "0")

  Data is output from the SCxTXD pin and the clock is output from the SCxSCLK 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")

  When data is written to the transmit buffer and the shift register is empty, or when data transmission from the shift register is completed, data is transferred to the shift register from the transmit buffer. Simultaneously, SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the transmit buffer has no data to be moved to the transmit shift register, INTTXx interrupt is not generated and the clock output stops.


Transmit data write timing
SCxSCLK output (SCxCR<SCLKS>="0" Rising edge mode)
SCxSCLK output (SCxCR<SCLKS>="1" Falling edge mode)
SCxTXD
bit 0
bit 1
bit 6
bit 7
bit 0
INTTXx interrupt request
SCxCR<TBEMP>

<WBUF> = "0" (if double buffering is disabled) (SCxCR<TIDLE>="10")

<WBUF> = "1" (if double buffering is enabled and there is data in buffer)

<WBUF> = "1" (if double buffering is enabled and there is no data in buffer) (SCxCR<TIDLE>="01")

Figure 12-15 Transmit Operation in the I/O Interface Mode (Clock Output Mode)

(2) Clock Input Mode

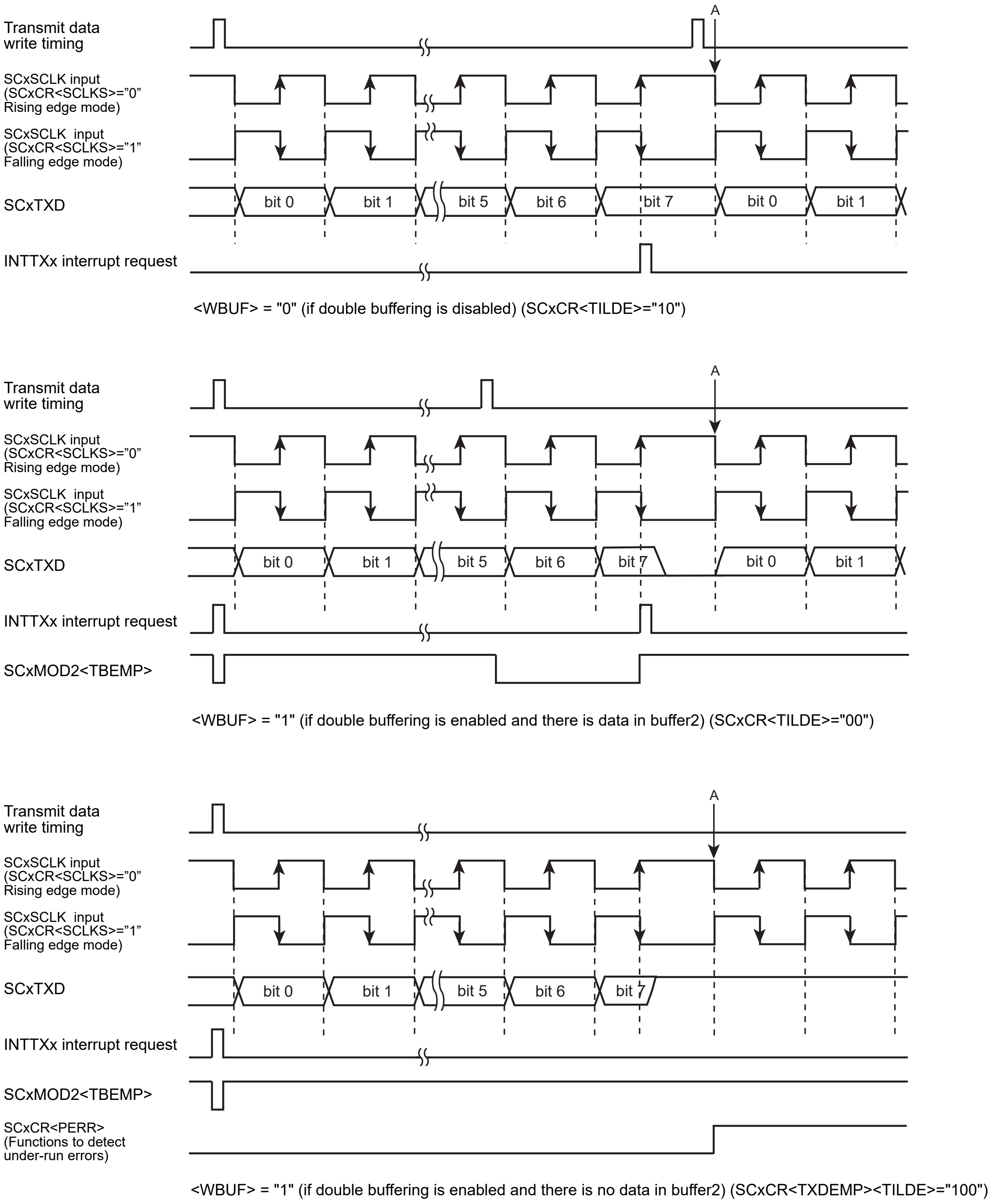
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  If the clock is input in the condition where data is written in the transmit buffer, 8-bit data is output from the SCxTXD pin. When all data is output, an interrupt INTTXx is generated. The next transmit data must be written before the timing of point "A" as shown in Figure 12-16.

- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  Data is moved from the transmit buffer to the transmit shift register when the CPU writes data to the transmit buffer before the clock input becomes active or when data transmission from the transmit shift register is completed. Simultaneously, SCxMOD2<TBEMP> is set to "1", and the INTTXx interrupt is generated.

  If the clock input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and the level which is specified by SCxCR<TXDEMP> is output to SCxTXD pin.


Transmit data write timing
A
SCxSCLK input (SCxCR<SCLKS>="0" Rising edge mode)
SCxSCLK input (SCxCR<SCLKS>="1" Falling edge mode)
SCxTXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
bit 1
INTTXx interrupt request
SCxMOD2<TBEMP>
SCxCR<PERR> (Functions to detect under-run errors)

<WBUF> = "0" (if double buffering is disabled) (SCxCR<TILDE>="10")

<WBUF> = "1" (if double buffering is enabled and there is data in buffer2) (SCxCR<TILDE>="00")

<WBUF> = "1" (if double buffering is enabled and there is no data in buffer2) (SCxCR<TXDEMP><TILDE>="100")

Figure 12-16 Transmit Operation in the I/O Interface Mode (Clock Input Mode)

12.16.1.2 Receive

(1) Clock Output Mode

The clock output starts by setting the receive enable bit SCxMOD0<RXE> to "1".

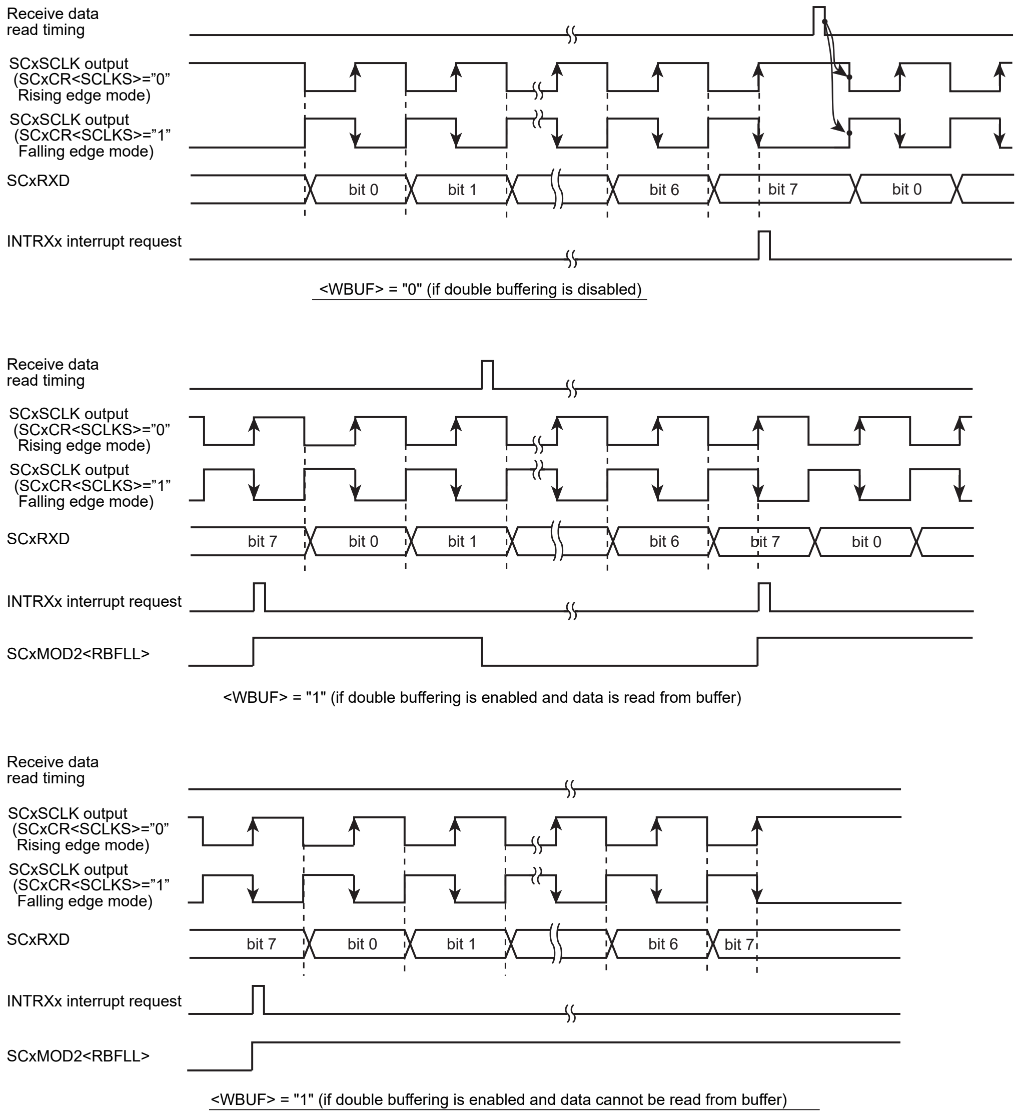
- If double buffer is disabled (SCxMOD2<WBUF> = "0")

  A clock is output from the SCxSCLK 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, SCxMOD2<RBFLL> is set to "1" and the INTRXx is generated.

  When a data is in the receive buffer, if the data is not read from the receive buffer before completing reception of the next 8 bits, the INTRXx interrupt is not generated and the clock 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.


Receive data read timing
SCxSCLK output (SCxCR<SCLKS>="0" Rising edge mode)
SCxSCLK output (SCxCR<SCLKS>="1" Falling edge mode)
SCxRXD
bit 0
bit 1
bit 6
bit 7
bit 0
INTRXx interrupt request
Receive data read timing
SCxSCLK output (SCxCR<SCLKS>="0" Rising edge mode)
SCxSCLK output (SCxCR<SCLKS>="1" Falling edge mode)
SCxRXD
bit 7
bit 0
bit 1
bit 6
bit 7
bit 0
INTRXx interrupt request
SCxMOD2<RBFLL>
Receive data read timing
SCxSCLK output (SCxCR<SCLKS>="0" Rising edge mode)
SCxSCLK output (SCxCR<SCLKS>="1" Falling edge mode)
SCxRXD
bit 7
bit 0
bit 1
bit 6
bit 7
INTRXx interrupt request
SCxMOD2<RBFLL>

<WBUF> = "0" (if double buffering is disabled)

<WBUF> = "1" (if double buffering is enabled and data is read from buffer)

<WBUF> = "1" (if double buffering is enabled and data cannot be read from buffer)

Figure 12-17 Receive Operation in the I/O Interface Mode (Clock Output Mode)

(2) clock input mode

In the clock input mode, receiving double buffering is always enabled, the received data can be moved to the receive buffer from the shift register, and the receive buffer can receive the next data successively.

The INTRXx receive interrupt is generated each time received data is moved to the receive buffer.

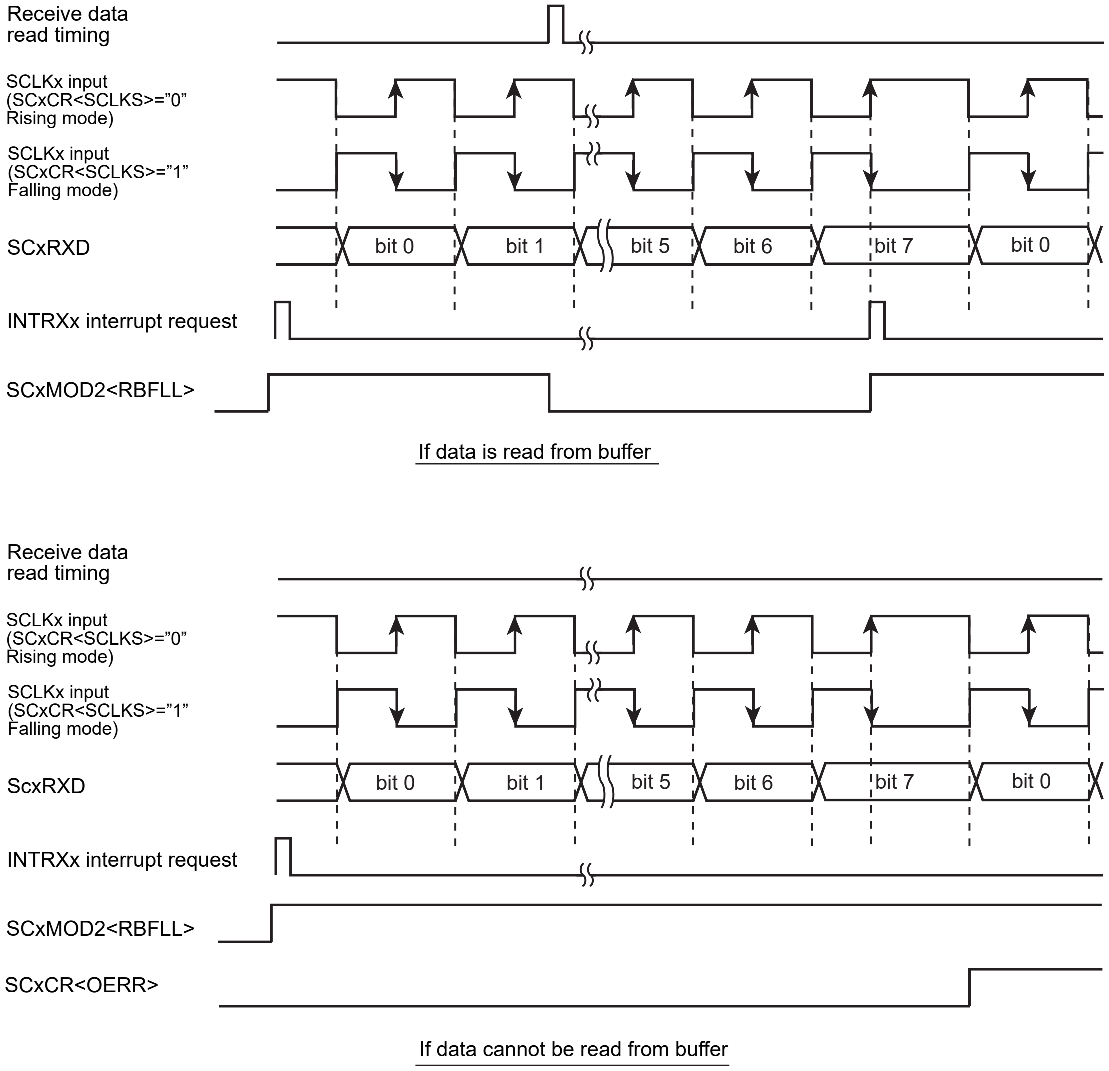

Receive data read timing
SCLKx input (SCxCR<SCLKS>="0" Rising mode)
SCLKx input (SCxCR<SCLKS>="1" Falling mode)
SCxRXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
INTRXx interrupt request
SCxMOD2<RBFLL>
If data is read from buffer
Receive data read timing
SCLKx input (SCxCR<SCLKS>="0" Rising mode)
SCLKx input (SCxCR<SCLKS>="1" Falling mode)
ScxRXD
bit 0
bit 1
bit 5
bit 6
bit 7
bit 0
INTRXx interrupt request
SCxMOD2<RBFLL>
SCxCR<OERR>
If data cannot be read from buffer

Figure 12-18 Receive Operation in the I/O Interface Mode (Clock Input Mode)

12.16.1.3 Transmit and Receive (Full-duplex)

(1) Clock Output Mode

- If double buffers are disabled (SCxMOD2<WBUF> = "0")

  Clock is output when the CPU writes data to the transmit buffer.

  Subsequently, a data is shifted into receive buffer and the INTRXx is generated. Concurrently, a data written to the transmit buffer is output from the SCxTXD pin, the INTTXx is generated when transmission of all data has been completed. Then, the clock output stops.

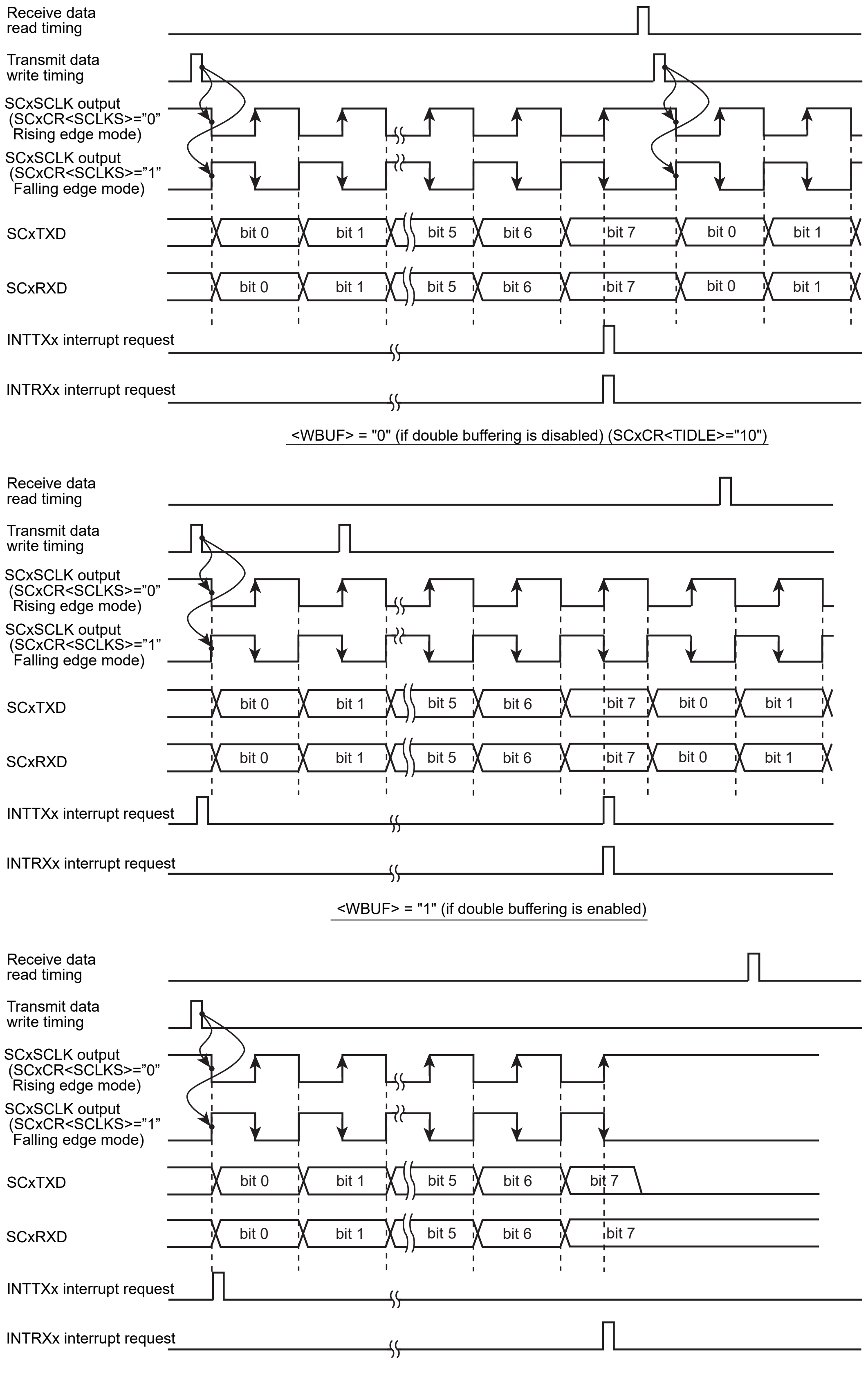
  The next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

- If double buffers are enabled (SCxMOD2<WBUF> = "1")

  Clock is outputted when the CPU writes data to the transmit buffer.

  A data is shifted into the receive shift register, moved to the receive buffer, and the INTRXx is generated. While a data is received, a transmit data is output from the SCxTXD pin. When all data are sent out, the INTTXx is generated and the next data is moved from the transmit buffer to the transmit shift register.

  If the transmit buffer has no data to be moved to the transmit buffer (SCxMOD2<TBEMP> = "1") or when the receive buffer is full (SCxMOD2<RBFLL> = "1"), the clock output stops. When both conditions, receive data is read and transmit data is written, are satisfied, the clock output is resumed and the next round of data transmission and reception is started.



<WBUF> = "0" (if double buffering is disabled) (SCxCR<TIDLE>="10")

<WBUF> = "1" (if double buffering is enabled)

<WBUF> = "1" (if double buffering is enabled) (SCxCR<TIDLE>="00")

Figure 12-19 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(2) Clock Input Mode

- If double buffers are disabled. (SCxMOD2<WBUF> = "0")

  When receiving data, double buffer is always enabled regardless of the SCxMOD2 <WBUF> settings.

  A data written in the transmit buffer is outputted from the SCxTXD pin and a data is shifted into the receive buffer when the clock input becomes active. The INTTXx is generated upon completion of data transmission. The INTRXx is generated when the data is moved from shift register to receive buffer after completion of data reception.

  Note that transmit data must be written into the transmit buffer before the clock input for the next data (data must be written before the point A in Figure 12-20). Data must be read before completing reception of the next data.

- If double buffers are enabled. (SCxMOD2<WBUF> = "1")

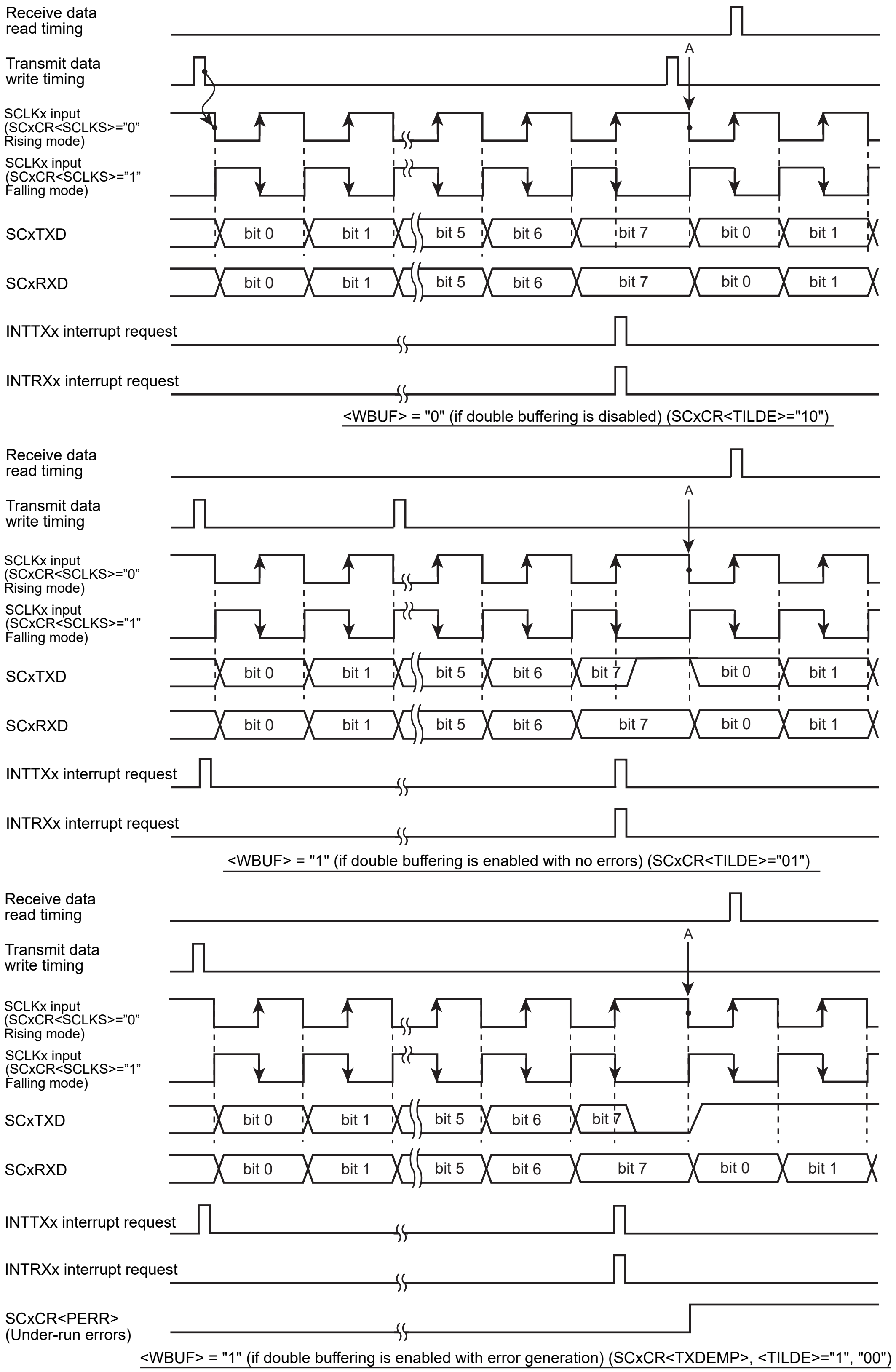
  The INTTXx is generated at the timing the transmit buffer data is moved to the transmit shift register after completing data transmission from the transmit shift register. At the same time, data received is shifted to the shift register, it is moved to the receive buffer, and the INTRXx is generated.

  Note that transmit data must be written into the transmit buffer before the clock input for the next data (data must be written before the point A in Figure 12-20). Data must be read before completing reception of the next data.

  Upon the clock input for the next data, 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 data is received, an overrun error occurs.

  If there is no data written to transmit buffer when clock for the next data is input, an underrun error occurs. The level which is specified by SCxCR<TXDEMP> is output to SCxTXD pin.



<WBUF> = "0" (if double buffering is disabled) (SCxCR<TILDE>="10")

<WBUF> = "1" (if double buffering is enabled with no errors) (SCxCR<TILDE>="01")

<WBUF> = "1" (if double buffering is enabled with error generation) (SCxCR<TXDEMP>, <TILDE>="1", "00")

Figure 12-20 Transmit/Receive Operation in the I/O Interface Mode (Clock Input Mode)

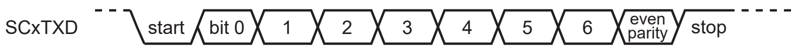
12.16.2 Mode 1 (7-bit UART mode)

The 7-bit UART mode is selected by setting SCxMOD<SM[1:0]> to "01".

In this mode, parity bits can be added to the transmit data stream; 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>. 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.



(Transmission rate of 2400 bps @fsys = 9.8304 MHz)

Clocking conditions	system clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

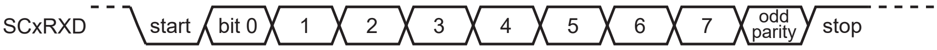
		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	-	0	0	1	0	1	Set 7-bit UART mode
SCxCR	←	x	1	1	x	x	x	0	0	Even parity enabled
SCxBRCR	←	0	0	1	0	0	1	0	0	Set 2400bps
SCxBUF	←	*	*	*	*	*	*	*	*	Set transmit data

x: don't care - : no change

12.16.3 Mode 2 (8-bit UART mode)

The 8-bit UART mode is 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:



(Reception rate of 9600 bps @fc = 9.8304 MHz)

Clocking conditions	System clock:	High-speed (fc)
	High-speed clock gear:	x 1 (fc)
	Prescaler clock:	fperiph/2 (fperiph = fsys)

		7	6	5	4	3	2	1	0	
SCxMOD0	←	x	0	0	0	1	0	0	1	Set 8-bit UART mode
SCxCR	←	x	0	1	x	x	x	0	0	Odd parity enabled
SCxBRCR	←	0	0	0	1	0	1	0	0	Set 9600bps
SCxMOD0	←	-	-	1	-	-	-	-	-	Reception enabled

x: don’t care - : no change

12.16.4 Mode 3 (9-bit UART mode)

The 9-bit UART mode is selected by setting SCxMOD0<SM[1:0]> to "11". In this mode, parity bits must be disabled (SCxCR<PE> = "0").

The most significant bit (9th bit) is written to SCxMOD0<TB8> for transmitting data. The data is stored in SCxCR<RB8> for receiving data.

When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SCxBUF.

The stop bit length can be specified using SCxMOD2<SBLEN>.

12.16.4.1 Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting SCxMOD0<WU> to "1".

In this case, the interrupt INTRXx will be generated only when SCxCR<RB8> is set to "1".

Note: The SCxTXD pin of the slave controller must be set to the open drain output mode using the PxOD.

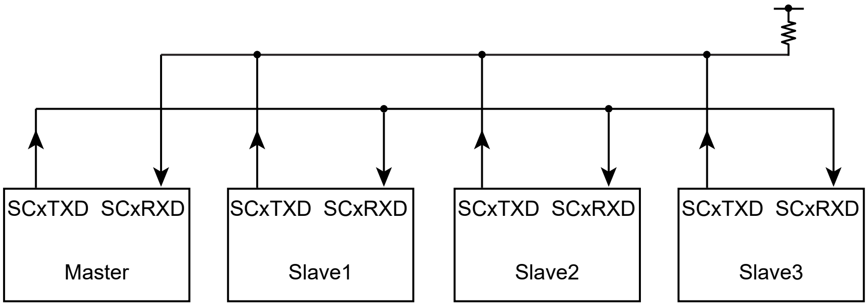


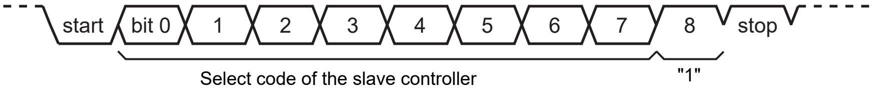
Figure 12-21 Serial Links to Use Wake-up Function

12.16.4.2 Protocol

1. Select the 9-bit UART mode for the master and slave controllers.

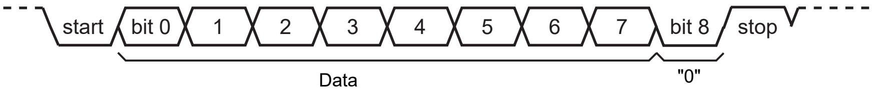
2. Set SCxMOD<WU> to "1" for the slave controllers to make them ready to receive data.

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> to "0".

5. The master controller transmits data to the designated slave controller (the controller of which SCxMOD<WU> 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> 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> set to "0" can transmit data to the master controller to inform that the data has been successfully received.

13. Serial Bus Interface (I2C/SIO)

The TMPM37AFSQG contains Serial Bus Interface (I2C/SIO), in which the following two operating modes are included:

- I2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

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

13.1 Configuration

The configuration of Serial bus interface is shown in Figure 13-1.

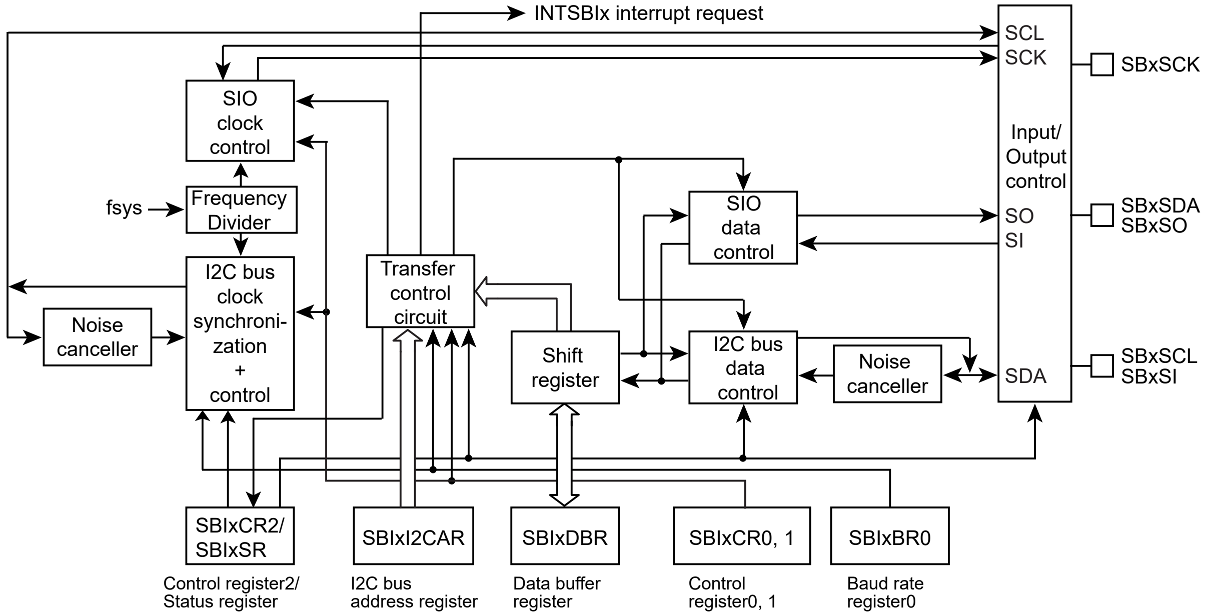


Figure 13-1 Serial Bus Interface Block Diagram

13.2 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 "13.3.1 Control Registers in the I2C Bus Mode" and "13.4.1 Control register of SIO mode".

13.2.1 Registers for each channel

The table below show control registers and their addresses.

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

Register name		Address(Base+)
Control register 0	SBIxCR0	0x0000
Control register 1	SBIxCR1	0x0004
Data buffer register	SBIxDBR	0x0008
I2C bus address register	SBIxI2CAR	0x000C
Control register 2	SBIxCR2 (writing)	0x0010
Status register	SBIxSR (reading)	
Baud rate register 0	SBIxBR0	0x0014

13.3 I2C Bus Mode

13.3.1 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.

13.3.1.1 SBIxCR0(Control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation 0:Disable 1:Enable To use the serial bus interface, enable this bit first. For the first time in case of setting to enable, the relevant SBI registers can be read or written. Since all clocks except SBIxCR0 stop if this bit is disabled, power consumption can be reduced by disabling this bit. If this bit is disabled after it’s been enabled once, the settings of each register are retained.
6-0	-	R	Read as 0.

13.3.1.2 SBIxCR1(Control register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
After reset	0	0	0	0	1	0	0	1(Note3)

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Function</th></tr>
<tr><td>31-8</td><td>-</td><td>R</td><td>Read as 0.</td></tr>
<tr><td>7-5</td><td>BC[2:0]</td><td>R/W</td><td>Select the number of bits per transfer (Note 1)
<table>
<tr><th rowspan="2">&lt;BC&gt;</th><th colspan="2">When &lt;ACK&gt; = 0</th><th colspan="2">When &lt;ACK&gt; = 1</th></tr>
<tr><th>Number of clock cycles</th><th>Data length</th><th>Number of clock cycles</th><th>Data length</th></tr>
<tr><td>000</td><td>8</td><td>8</td><td>9</td><td>8</td></tr>
<tr><td>001</td><td>1</td><td>1</td><td>2</td><td>1</td></tr>
<tr><td>010</td><td>2</td><td>2</td><td>3</td><td>2</td></tr>
<tr><td>011</td><td>3</td><td>3</td><td>4</td><td>3</td></tr>
<tr><td>100</td><td>4</td><td>4</td><td>5</td><td>4</td></tr>
<tr><td>101</td><td>5</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>110</td><td>6</td><td>6</td><td>7</td><td>6</td></tr>
<tr><td>111</td><td>7</td><td>7</td><td>8</td><td>7</td></tr>
</table></td></tr>
<tr><td>4</td><td>ACK</td><td>R/W</td><td>Master mode
0: Acknowledgement clock pulse is not generated.
1: Acknowledgement clock pulse is generated.
Slave mode
0: Acknowledgement clock pulse is not counted.
1: Acknowledgement clock pulse is counted.</td></tr>
<tr><td>3</td><td>-</td><td>R</td><td>Read as 1.</td></tr>
<tr><td>2-1</td><td>SCK[2:1]</td><td>R/W</td><td rowspan="2">Select internal SCL output clock frequency (Note 2).
<table>
<tr><td>000</td><td>n = 5</td><td rowspan="8">System Clock : fsys
Clock gear : fc/1
Frequency = $\frac{fsys}{2^n + 72}$ [Hz]</td></tr>
<tr><td>001</td><td>n = 6</td></tr>
<tr><td>010</td><td>n = 7</td></tr>
<tr><td>011</td><td>n = 8</td></tr>
<tr><td>100</td><td>n = 9</td></tr>
<tr><td>101</td><td>n = 10</td></tr>
<tr><td>110</td><td>n = 11</td></tr>
<tr><td>111</td><td>reserved</td></tr>
</table></td></tr>
<tr><td rowspan="2">0</td><td>SCK[0]</td><td>W</td></tr>
<tr><td>SWRMON</td><td>R</td><td>On reading <SWRMON>: Software reset status monitor
0:Software reset operation is in progress.
1:Software reset operation is not in progress.</td></tr>
</table>

Note 1: Clear <BC[2:0]> to "000" before switching the operation mode to the SIO mode.

Note 2: For details on the SCL line clock frequency, refer to "13.3.2.2 Serial Clock"

Note 3: After a reset, the <SCK[0]/SWRMON> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value of the <SCK[0]> bit is "0".

Note 4: The initial value for selecting a frequency is <SCK[2:0]>=000 and is independent of the read initial value.

Note 5: When <BC[2:0]>="001" and <ACK>="0" in master mode, SCL line may be fixed to "L" by falling edge of SCL line after generation of STOP condition and the other master devices cannot use the bus. In the case of bus which is connected with several master devices, the number of bits per transfer should be set equal or more than 2 before generation of STOP condition.

13.3.1.3 SBIxCR2(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	MST	TRX	BB	PIN	SBIM		SWRST	
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note1) 00: Port mode (Disables a serial bus interface output) (Note2) 01: SIO mode 10: I2C bus mode (Note3) 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset. For details, refer to "13.3.2.16 Software Reset".

Note 1: Make sure that modes are not changed during a communication session.

Note 2: Ensure that the bus is free before switching the operating mode to the port mode.

Note 3: Ensure that the SBxSDA pin and SBxSCL pin are at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.

Note 4: SBIxCR2 is assigned at same address with SBIxSR. Thus, read-modify-write operation cannot be used.

13.3.1.4 SBIxSR (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	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	R	Master/slave selection monitor 0: Slave mode 1: Master mode
6	TRX	R	Transmit/receive selection monitor 0: Receive 1: Transmit
5	BB	R	I2C bus state monitor 0: Free 1: Busy
4	PIN	R	INTSBIx interrupt request monitor 0:Interrupt request generated 1: Interrupt request cleared
3	AL	R	Arbitration lost detection 0: − 1:Detected
2	AAS	R	Slave address match detection 0: − 1: Detected (This bit is set when the general-call address is detected as well.)
1	AD0	R	General call detection 0: − 1:Detected
0	LRB	R	Last received bit monitor 0:Last received bit "0" 1:Last received bit "1"

13.3.1.5 SBIxBR0(Serial bus interface baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation at the IDLE mode 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Be sure to write "0".

13.3.1.6 SBIxDBR (Serial bus interface data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive 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 SBIxBDR 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.

13.3.1.7 SBIxI2CAR (I2Cbus address register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-1	SA[6:0]	R/W	Set the slave address when the SBI acts as a slave device.
0	ALS	R/W	Specify address recognition mode. 0: Recognize its slave address. 1: Do not recognize its slave address (free-data format).

Note 1: Please set the bit 0 <ALS> of I2C bus address register SBIxI2CAR to "0", except when you use a free data format. It operates as a free data format when setting it to "1". Selecting the master fixes to transmission. Selecting the slave fixes to reception.

Note 2: Do not set SBIxI2CAR to "0x00" in slave mode. (If SBIxI2CAR is set to "0x00", it’s recognized that the slave address matches the START byte ("0x01") of the I2C standard received in slave mode.)

13.3.2 Control

13.3.2.1 Operating mode

The setting of SBIxCR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, set <SBIM [1:0]> to "10".

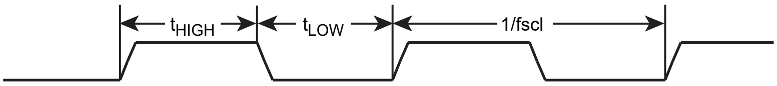
Note 1: Ensure that the SBxSDA pin and SBxSCL pin are at the "High" level before changing to the port mode.

Note 2: Ensure that the SBxSDA pin and SBxSCL pin are at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.

13.3.2.2 Serial Clock

(1) Clock source

SBIxCR1<SCK[2:0]> specifies the maximum frequency of the serial clock to be output from the SBxSCL pin in the master mode.



$$t_{LOW} = 2^{n-1}/fsys + 58/fsys$$
$$t_{HIGH} = 2^{n-1}/fsys + 14/fsys$$
$$fscl = 1/(t_{LOW} + t_{HIGH}) = \frac{fsys}{2^n + 72}$$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 13-2 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.

(2) Clock Synchronization

The I2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "Low" level overrides other masters producing the "High" level on their clock lines. This must be detected and responded by the masters producing the "High" level.

Clock synchronization assures correct data transfer on a bus that has two or more master.

For example, the clock synchronization procedure for a bus with two masters is shown below.

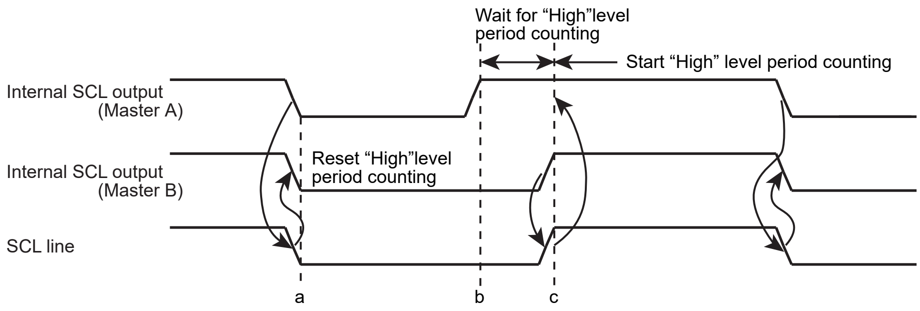

Wait for "High"level period counting
Start "High" level period counting
Internal SCL output (Master A)
Reset "High"level period counting
Internal SCL output (Master B)
SCL line
a
b
c

Figure 13-3 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 SBxSCL 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.

13.3.2.3 Setting the Acknowledgement Mode

Setting SBIxCR1<ACK> to "1" selects the acknowledge mode.

When operating as a master, the SBI adds one clock for acknowledgment signal.

In slave mode, the clock for acknowledgement signals is counted.

In transmitter mode, the SBI releases the SBxSDA pin during clock cycle to receive acknowledgement signals from the receiver.

In receiver mode, the SBI pulls the SBxSDA 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 SBxSDA pin to the "Low" level during the clock cycle and generates acknowledgement signals. However, the second byte of the general call is necessary to be controlled by software to generate an acknowledgement signal depending on the contents of the second byte.

By setting <ACK> to "0", the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals. In slave mode, the clock for acknowledgement signals is counted.

13.3.2.4 Setting the Number of Bits per Transfer

SBIxCR1<BC[2:0]> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC[2:0]> is set to "000", causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC[2:0]> keeps a previously programmed value.

13.3.2.5 Slave Addressing and Address Recognition Mode

Setting "0" to SBIxI2CAR<ALS> and a slave address in SBIxI2CAR<SA[6:0]> sets addressing format, and then the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format.

If <ALS> is set to "1", the SBI does not recognize a slave address and receives data in the free data format. In the case of free data format, a slave address and a direction bit are not recognized; they are recognized as data immediately after generation of the start condition.

13.3.2.6 Configuring the SBI as a Master or a Slave

Setting SBIxCR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device.

<MST> is cleared to "0" by the hardware when it detects the stop condition on the bus or the arbitration lost.

13.3.2.7 Configuring the SBI as a Transmitter or a Receiver

Setting SBIxCR2<TRX> to "1" configures the SBI as a transmitter.

Setting <TRX> to "0" configures the SBI as a receiver.

If SBI is used in free data format, <TRX> is not changed by the hardware.

If SBI is used in addressing format, <TRX> is set shown as follow.

(1) Master mode

As a master mode, if SBI receives acknowledgement from a slave device, <TRX> is set shown as below by a hardware.

If SBI does not acknowledgement, <TRX> retains the previous value.

- When the transmitted direction bit is "1", <TRX> is set to "0"
- When the transmitted direction bit is "0", <TRX> is set to "1".

(2) Slave mode

As a slave mode, in case of addressing format, if below condition is satisfied, <TRX> is set depended on the direction bit which is sent by a master device.

- When the received slave address is as same as the value set in SBIxI2CAR.
- When SBI receives general-call

<TRX> is set shown as below.

- When the received direction bit is "1", <TRX> is set to "0".
- When the received direction bit is "0", <TRX> is set to "1".

13.3.2.8 Bus busy monitor

To conform the state of the bus, read SBIxSR<BB>.

<BB> is set to "1" when SBI detects the start condition on the bus and is cleared to "0" when SBI detects the stop condition on the bus.

When <BB> is "1", it is called as bus busy. When <BB> is "0", it is called as bus free.

The master device can generate the start condition in only bus free. It should be conform that <BB> is "0".

When <BB> is "1", SBI generates the start condition, the start condition is not generated and the arbitration lost is occurred.

13.3.2.9 Interrupt Service Request and Release

When INTSBIx is generated, SBIxCR2<PIN> is cleared to "0" and SBI is in interrupt service request state. SBI pulls SBxSCL pin to "Low" level during <PIN> is "0".

<PIN> is set to "1" when data is written to or read from SBIxDBR. When the program writes "1" to <PIN>, it is set to "1". However, writing "0" does not clear <PIN> to "0".

If <PIN> is set to "1", SBxSCL pin is released. It takes tLOW from setting <PIN> to "1" to releasing SBxSCL pin.

Note: When arbitration occurs while a slave address and direction bit are transferred in the master mode, <PIN> is cleared to "0" and INTSBIx occurs. This does not relate to whether a slave address matches <SA>.

13.3.2.10 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.

The I2C-bus arbitration takes place on the SDA bus 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 SBxSDA pin, so that it does not affect the data transfer initiated by another master.

If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

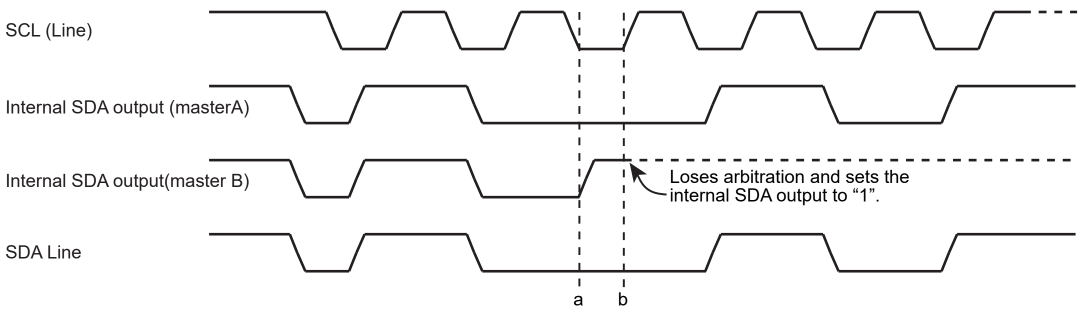

SCL (Line)
Internal SDA output (masterA)
Internal SDA output(master B)
SDA Line
Loses arbitration and sets the internal SDA output to “1”.
a
b

Figure 13-4 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, Arbitration Lost occurs and SBIxSR<AL> is set to "1".

When an arbitration lost occurs, SBIxSR<MST> and <TRX> are cleared to "0", causing the SBI to operate as a slave receiver and it stops the SCL clock output during data transfer.

The device which generates the arbitration lost in the transferring a slave address receives a slave address which is transmitted by other master devices as like as a slave device.

When the received slave address is matched with SBIxI2CAR<SA>, <PIN> is cleared to "0" and INTI2Cx is occurred when it is not matched, <PIN> is remains "1" and INTxSBI is occurred.

<AL> is cleared to "0" when data is written to or read from SBIxDBR or data is written to SBIxCR2.

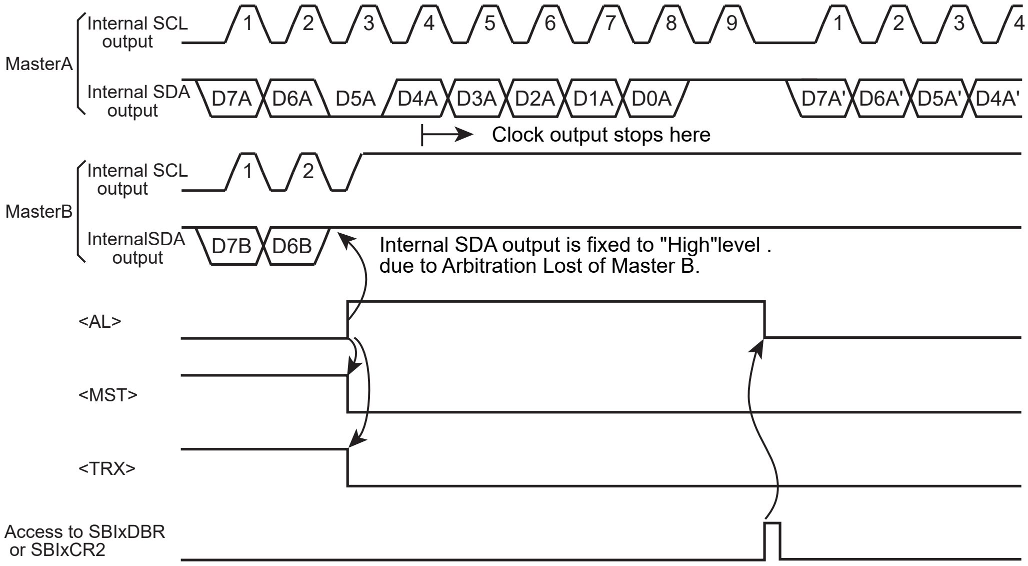


Figure 13-5 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

13.3.2.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBIxI2CAR<ALS>="0"), SBIxSR<AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at SBIxI2CAR.

When <ALS> is "1", <AAS> is set to "1" when the first data word has been received.

<AAS> is cleared to "0" when data is written to or read from SBIxDBR.

13.3.2.12 General-call Detection Monitor

When the SBI operates as a slave device, SBIxSR<AD0> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeroes.

<AD0> is cleared to "0" when the start or stop condition is detected on the bus.

13.3.2.13 Last Received Bit Monitor

SBIxSR<LRB> is set to the SDA line value that was read at the rising of the SCL line.

In the acknowledgment mode, reading SBIxSR<LRB> immediately after generation of the INTSBIx interrupt request causes ACK signal to be read.

13.3.2.14 Data Buffer Register (SBIxDBR)

Reading or writing SBIxDBR initiates reading received data or writing transmitted data.

When the SBI is in the master mode, after writing a slave address and a direction bit to this register in SBIxDBR, the start condition is generated, SBI transmits a slave address and a direction bit to slave device.

13.3.2.15 Baud Rate Register (SBIxBR0)

The SBIxBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode.

This register must be programmed before executing an instruction to switch to the standby mode.

13.3.2.16 Software Reset

If SBI locks up due to external noise, it can be initialized by using a software reset.

In I2C bus mode, writing "10" followed by "01" to SBIxCR2<SWRST[1:0]> generates a reset signal that initializes SBI. When writing SBIxCR2<SWRST[1:0]>, set SBIxCR2<SBIM[1:0]> to "10" for I2C bus mode. After a reset, all control registers and status flags are initialized to their reset values. When SBI is initialized, <SWRST> is automatically cleared to "00".

Note: A software reset causes the SBI operating mode to switch from the I2C mode to the port mode.

13.3.3 Data Transfer Procedure

13.3.3.1 Device Initialization

Firstly, set SBIxCR1<ACK><SCK[2:0]>. Set "1" to <ACK> to specify the acknowledgement mode. Set "000" to SBIxCR1<BC[2:0]> .

Secondly, set <SA[6:0]> (a slave address) and <ALS> to SBIxI2CAR . (In the addressing format mode, set <ALS>="0").

Finally, to configure the Serial Bus Interface as a slave receiver, ensure that the serial bus interface pin is at "High" first. Then write "000" to SBIxCR2<MST><TRX><BB>, "1" to <PIN>, "10" to <SBIM[1:0] > and "00" to <SWRST[1:0]>.

Note: Initialization of the serial bus interface circuit must be completed within a period that any device does not generate start condition after all devices connected to the bus were initialized. If this rule is not followed, data may not be received correctly because other devices may start transfer before the initialization of the serial bus interface circuit is completed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	0	X	0	X	X	X	Specifies ACK and SCL clock.
SBIxI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBIxCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

Note: X: Don’t care

13.3.3.2 Generating the Start Condition and a Slave Address

The following steps are required to generate the start condition and slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBIxCR1<ACK> to select the acknowledgment mode. Write to SBIxDBR a slave address and a direction bit to be transmitted.

When <BB> = "0", writing "1111" to SBIxCR2<MST><TRX><BB><PIN> generates the start condition on the bus.

Following the start condition, the SBI generates nine clocks from the SBxSCL pin.

The SBI outputs the slave address and the direction bit specified at SBIxDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIx interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0".

The SBI holds the SCL line at the "Low" level while <PIN> is = "0".<TRX> changes its value according to the transmitted direction bit at generation of the INTSBIx interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Note: To output salve address, check with software that the bus is free before writing to SBIxDBR. If this rule is not followed, data being output on the bus may get ruined.

Settings in main routine

```
                      7   6   5   4   3   2   1   0
   Reg.          ←    SBIxSR
┌→ Reg.          ←    Reg.AND 0x20
│  if Reg.       ≠    0x00                               Ensures that the bus is free.
│
│  Then
└─
   SBIxCR1       ←    X   X   X   1   0   X   X   X      Selects the acknowledgement mode.
   SBIxDBR       ←    X   X   X   X   X   X   X   X      Specifies the desired slave address and direction.
   SBIxCR2       ←    1   1   1   1   0   0   0   0      Generates the start condition.
                 Example of INTSBIx interrupt routine
                      Clears the interrupt request.
                      Processing
                      End of interrupt
```

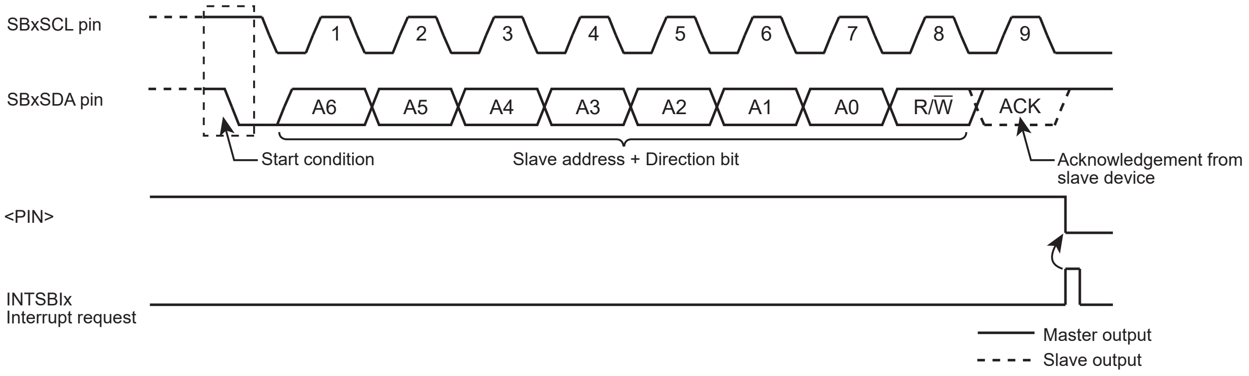


Figure 13-6 Generation of the Start Condition and a Slave Address

13.3.3.3 Transferring a Data Word

At the end of a data word transfer, the INTSBIx interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

(1) Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

(a) Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1", that means the receiver requires no further data. The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0", that means the receiver requires further data.

If the next data to be transmitted has eight bits, the data is written into SBIxDBR. If the data has different length, <BC[2:0]> and <ACK> are programmed and the transmit data is written into SBIxDBR.

Writing the data makes <PIN> to "1", causing the SBxSCL pin to generate a serial clock for transferring a next data word, and the SBxSDA pin to transfer the data word.

After the transfer is completed, the INTSBIx interrupt request is generated, <PIN> is cleared to "0", and the SBxSCL pin is pulled to the "Low" level.

To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIx interrupt

```
if MST = 0
Then go to the slave-mode processing.
if TRX = 0
Then go to the receiver-mode processing.
if LRB = 0
Then go to processing for generating the stop condition.
SBIxCR1    ←  X  X  X  X  0  X  X  X    Specifies the number of bits to be transmitted and
                                        specify whether ACK is required.
SBIxDBR    ←  X  X  X  X  X  X  X  X    Writes the transmit data.
End of interrupt processing.
```

Note: **X: Don’t care**

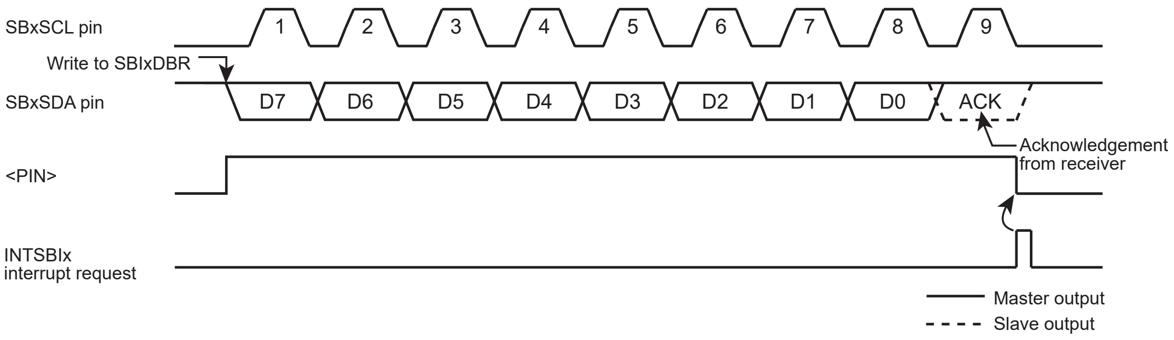


Figure 13-7 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(b) Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIxDBR. If the data has different length, <BC[2:0]> is programmed and the received data is read from SBIxDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.)

On reading the data, <PIN> is set to "1", and the serial clock is output to the SBxSCL pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the "Low" level, "0" is output to the SBxSDA pin.

After that, the INTSBIx interrupt request is generated, and <PIN> is cleared to "0", pulling the SBxSCL pin to the "Low" level. Each time the received data is read from SBIxDBR, one-word transfer clock and an acknowledgement signal are output.

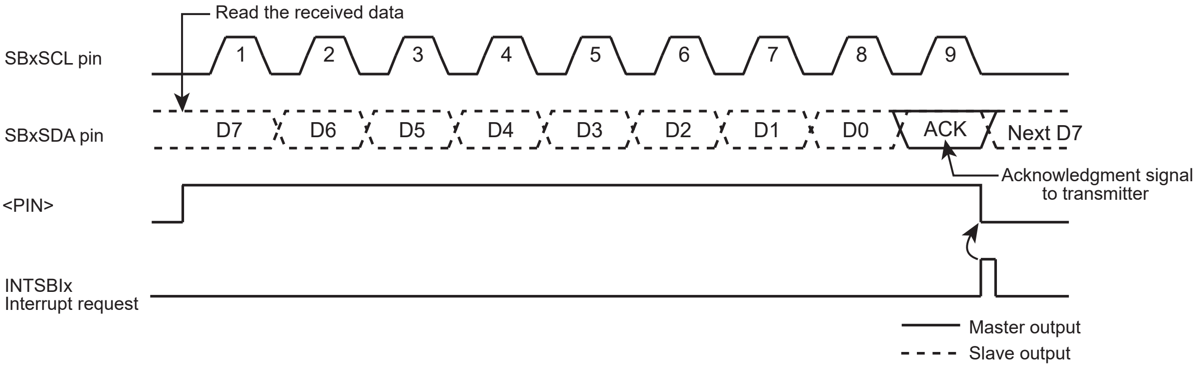


Figure 13-8 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

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.

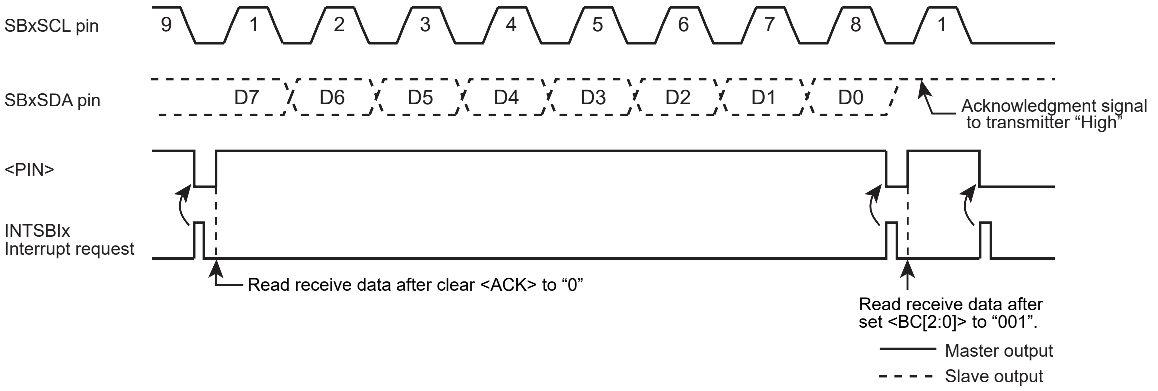

SBxSCL pin
9
1
2
3
4
5
6
7
8
1
SBxSDA pin
D7
D6
D5
D4
D3
D2
D1
D0
Acknowledgment signal to transmitter "High"
<PIN>
INTSBIx Interrupt request
Read receive data after clear <ACK> to "0"
Read receive data after set <BC[2:0]> to "001".
Master output
Slave output

Figure 13-9 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIx interrupt (after data transmission)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	X	0	X	X	X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	←	SBIxDBR								Reads dummy data.
End of interrupt										

INTSBIx interrupt (first to (N-2)th data reception)

		7	6	5	4	3	2	1	0	
Reg.	←	SBIxDBR								Reads the first to (N-2)th data words.
End of interrupt										

INTSBIx interrupt ((N-1)th data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	X	X	X	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the (N-1)th data word.
End of interrupt										

INTSBIx interrupt (Nth data reception)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	0	1	0	0	X	X	X	Disables generation of acknowledgement clock.
Reg.	←	SBIxDBR								Reads the Nth data word.
End of interrupt										

INTSBIx interrupt (after completing data reception)

Processing to generate the stop condition.	Terminates the data transmission.
End of interrupt	

Note: X: Don’t care

(2) Slave mode (<MST> = "0")

In the slave mode, SBI generates the INTSBIx interrupt request when SBI receives any slave address or general-call from master device, when SBI completes to transfers a data after SBI received its slave address or general-call. Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode. When the completion of data word transfer in which Arbitration Lost is detected, the INTSBIx interrupt request is generated. When INTSBIx interrupt request, <PIN> is cleared to "0", and SBxSCL pin is pulled to the "Low" level. When data is written to or read from SBIxDBR or when <PIN> is set to "1", SBxSCL pin is released after a period of t_{LOW}.

In addition, ACK signals are necessary to be controlled depending on the contents of the second byte by software.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out and it changes from master mode to slave mode.

SBIxSR<AL>, <TRX>, <AAS> and <AD0> are tested to determine the processing required.

"Table 13-1 Processing in Slave Mode" shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode.

INTSBIx interrupt

```
if TRX = 0
Then go to other processing.
if AL = 0
Then go to other processing.
if AAS = 0
Then go to other processing.
SBIxCR1    ←  X  X  X  1  0  X  X  X     Sets the number of bits to be transmitted.
SBIxDBR    ←  X  X  X  X  X  X  X  X     Sets the transmit data.
```

Note: X: Don’t care

Table 13-1 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration Lost is detected while the slave address was being transmitted and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC[2:0]> and write the transmit data into SBIxDBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1", that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0", that means the receiver requires further data. Set the number of bits in the data word to <BC[2:0]> and write the transmit data to the SBIxDBR.
0	1	1	1/0	Arbitration Lost is detected while a slave address is being transmitted, and the SBI receives either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIxDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration Lost is detected while a slave address or a data word is being transmitted, and the transfer is terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC [2:0].> and read the received data from SBIxDBR.

13.3.3.4 Generating the Stop Condition

When SBIxSR<BB> is "1", writing "1" to SBIxCR2<MST>, <TRX>, <PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus.

Do not alter the contents of <MST>, <TRX>, <BB>, <PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released.

After that, the SBxSDA pin goes "High", causing the stop condition to be generated.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

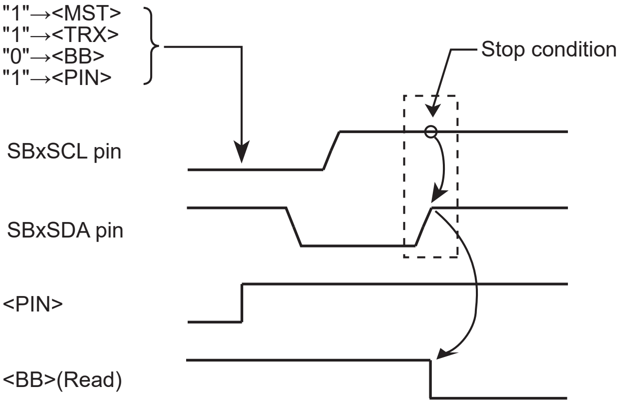


Figure 13-10 Generating the Stop Condition

13.3.3.5 Restart Procedure

Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a restart in the master mode is described below.

First, write SBIxCR2<MST>, <TRX>, <BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SBxSDA pin is held at the "High" level and the SBxSCL pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy. Then, test SBIxSR<BB> and wait until it becomes "0" to ensure that the SBxSCL pin is released. Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCL bus line to the "Low" level. Once the bus is determined to be free by following the above procedures, follow the procedures described in "13.3.3.2 Generating the Start Condition and a Slave Address" to generate the start condition.

To satisfy the setup time of restart, at least 4.7μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

Note 1: Do not write <MST> to "0" when it is "0". (Restart cannot be initiated.)

Note 2: When the master device is acting as a receiver, data transmission from the slave device which serves as a transmitter must be completed before generating a restart. To complete data transfer, slave device must receive a "High" level acknowledge signal. For this reason, <LBR> before generating a restart becomes "1", the rising edge of the SCL line is not detected even <LBR>= "1" is confirmed by following the restart procedure. To check the status of the SCL line, read the port.

		7	6	5	4	3	2	1	0	
SBIxCR2	←	0	0	0	1	1	0	0	0	Releases the bus.
if SBIxSR<BB> ≠ 0										Checks that the SBxSCL pin is released.
Then										
if SBIxSR<LRB> ≠ 1										Checks that no other device is pulling the SBxSCL pin to the "Low".
Then										
4.7 μs Wait										
SBIxCR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgment mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Sets the desired slave address and direction.
SBIxCR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Note: X: Don’t care

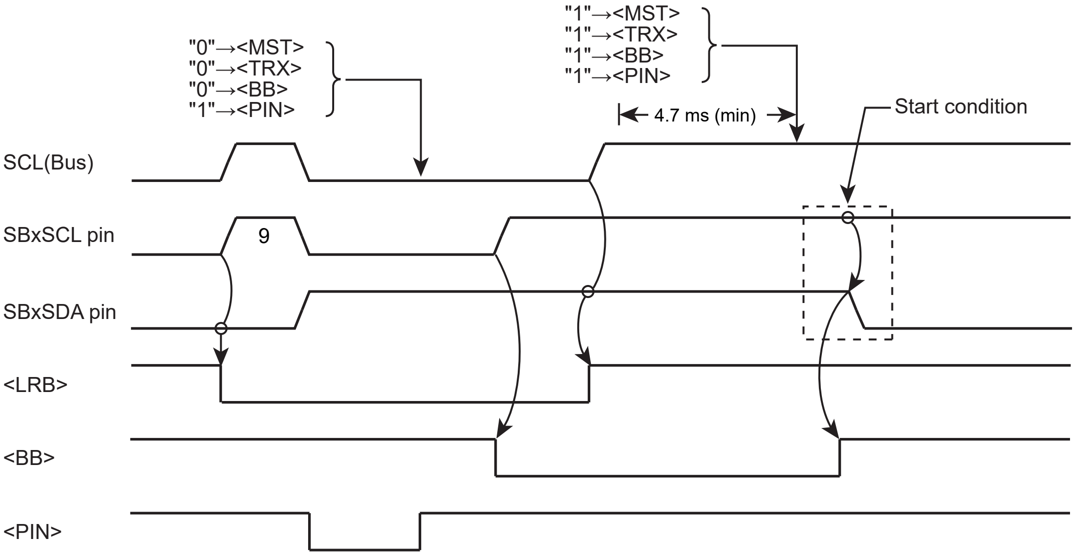


Figure 13-11 Timing Chart of Generating a Restart

13.3.4 Data Format

Figure 13-12 shows the data formats used in the I2C bus mode.

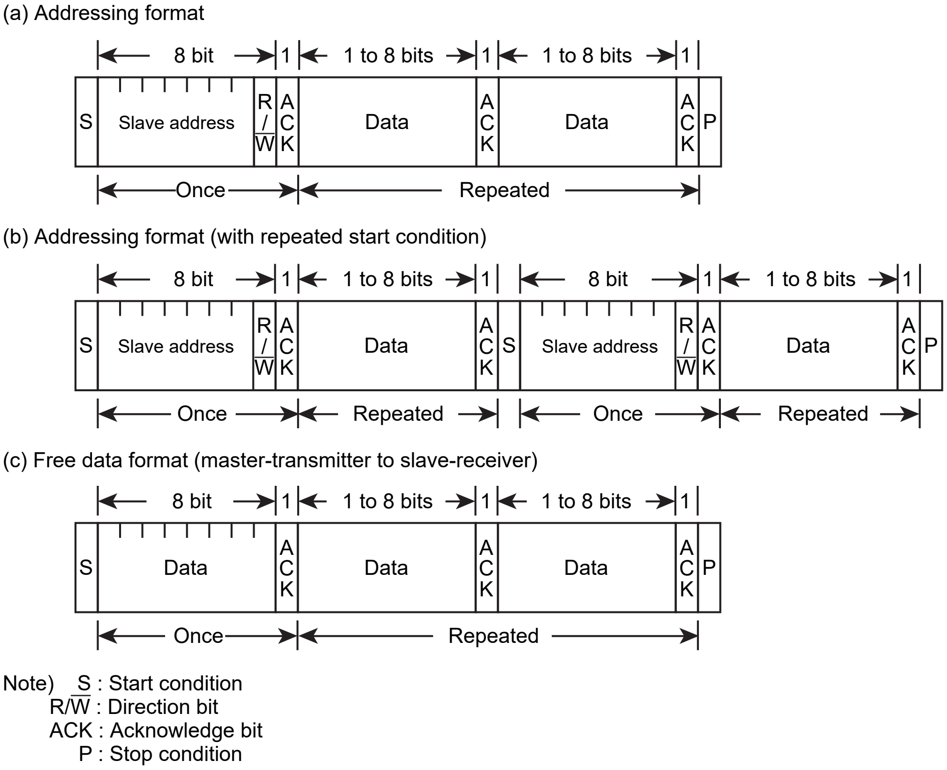


Figure 13-12 I2C Bus Mode Data Formats

13.3.5 Precautions on Use of Multi-master

Prepare recovery process by software in case that communication is in lock state in multi-master mode.

Example of recovery process

1. Start timer for timeout detection synchronizing with starting communication.
2. If a serial interface interrupt (INTSBIx) does not occur within the specified time, a timeout occurs and the MCU determines that communication is locked up.
3. Do software reset on serial bus interface to release the condition that communication is locked up.
4. Adjust transmission timings (note)
5. Resend transmission data.

Note: Adjust transmission timing between the MCUs to avoid overlapping the transmission timing.

13.4 SIO Mode

13.4.1 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.

13.4.1.1 SBIxCR0(control register 0)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SBIEN	R/W	Serial bus interface operation. 0:Disable 1: Enable Enable this bit before using the serial bus interface. If this bit is disabled, power consumption can be reduced because all clocks except SBIxCR0 stop. If the serial bus interface operation is enabled and then disabled, the settings will be maintained in each register.
6-0	-	R	Read as 0.

13.4.1.2 SBIxCR1(Control register 1)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	SIOS	R/W	Transfer Start/Stop 0: Stop 1: Start
6	SIOINH	R/W	Transfer 0: Continue 1: Forced termination
5-4	SIOM[1:0]	R/W	Select transfer mode 00: Transmit mode 01: Reserved 10:Transmit/receive mode 11:Receive mode
3	-	R	Read as 1.
2-0	SCK[2:0]	R/W	On writing <SCK[2:0]>: Select serial clock frequency. (Note 1) 000 n = 3 001 n = 4 010 n = 5 011 n = 6 100 n = 7 101 n = 8 110 n = 9 111 – System clock : fsys Clock gear : fc/1 Frequency = $\frac{fsys/2}{2^n}$ [Hz]

Note 1: After a reset, the <SCK[0]> bit is read as "1". However, if the SIO mode is selected at the SBIxCR2 register, the initial value is read as "0". In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state. The descriptions of the SBIxCR2 register and the SBIxSR register are the same.

Note 2: Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.

13.4.1.3 SBIxDBR (Data buffer register)

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-0	DB[7:0]	R	Receive data
		W	Transmit data

Note 1: The transmission data must be written in to the register from the MSB (bit 7). The received data is stored in the LSB.

Note 2: Since SBIxDBR has independent buffers for writing and reading, a written data cannot be read. Thus, read-modify-write instructions, such as bit manipulation, cannot be used.

13.4.1.4 SBIxCR2(Control register 2)

This register serves as SBIxSR register by writing to it.

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

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7-4	-	R	Read as 1. (Note 1)
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2Cbus mode 11: Reserved
1-0	-	R	Read as 1. (Note 1)

Note 1: In this document, the value written in the column "after reset" is the value after setting the SIO mode in the initial state.

Note 2: Make sure that modes are not changed during a communication session.

13.4.1.5 SBIxSR (Status Register)

This register serves as SBIxCR2 by writing to it.

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

13.4.1.6 SBIxBR0 (Baud rate register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
After reset	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	-	R	Read as 1.
6	I2SBI	R/W	Operation in IDLE mode. 0: Stop 1: Operate
5-1	-	R	Read as 1.
0	-	R/W	Make sure to write "0".

13.4.2 Control

13.4.2.1 Serial Clock

(1) Clock source

Internal or external clocks can be selected by programming SBIxCR1<SCK[2:0]>.

(a) 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 SBxSCK pin. At the beginning of a transfer, the SBxSCK 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.

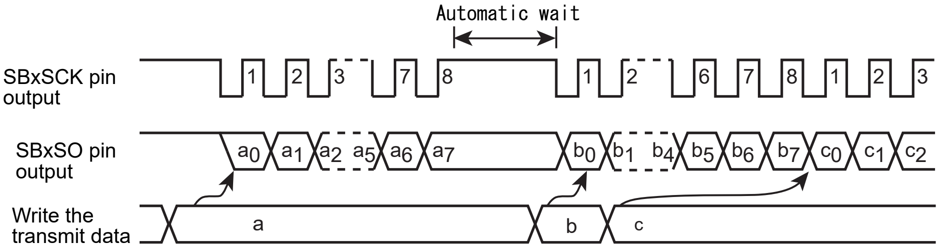
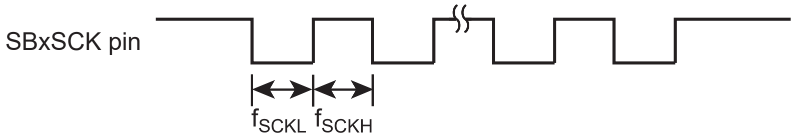


Figure 13-13 Automatic Wait

(b) External clock (<SCK[2:0]> = "111")

The SBI uses an external clock supplied from the outside to the SBxSCK 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 13-14 Maximum Transfer Frequency of External Clock Input

(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 SBxSCK pin input/output).

- Trailing-edge shift

  Data is shifted at the trailing edge of the serial clock (or the rising edge of the SBxSCK pin input/output).

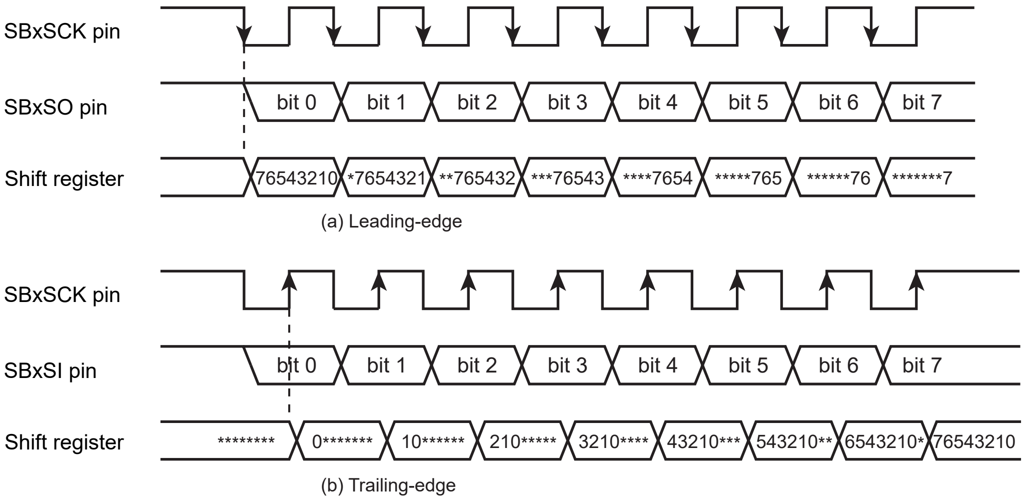


Figure 13-15 Shift Edge

13.4.2.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBIxCR1<SIOM[1:0]>.

(1) 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIxDBR.

After writing the transmit data, writing "1" to SBIxCR1<SIOS> starts the transmission. The transmit data is moved from SBIxDBR to a shift register and output to the SBxSO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIxDBR becomes empty, and the INTSBIx (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIxDBR is loaded with the next transmit data.

In the external clock mode, SBIxDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIxDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBIxSR<SIOF> to "1" to the falling edge of SCK line.

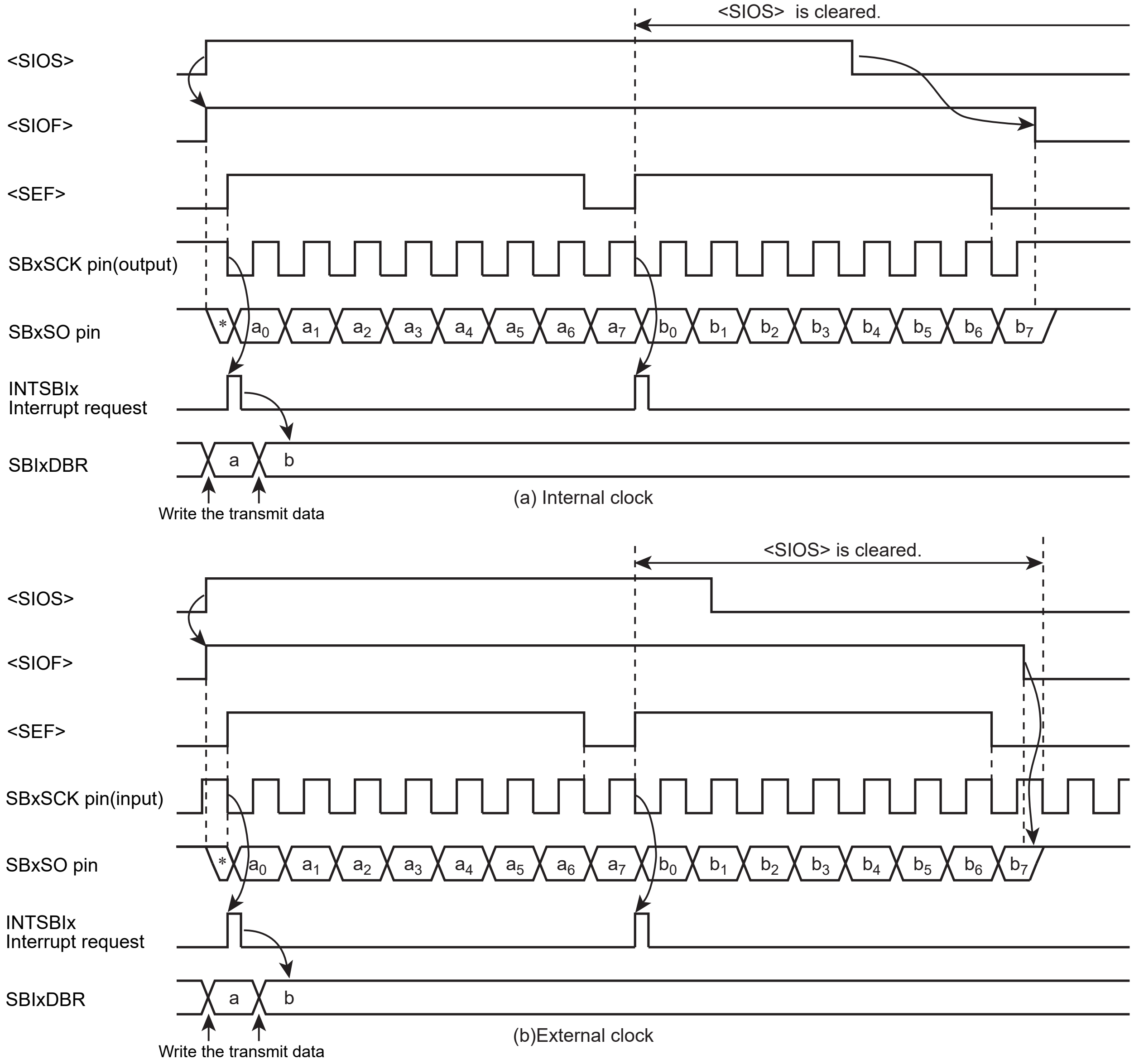
Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBIxSR<SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1", the transmission is aborted immediately and <SIOF> is cleared to "0".

When in the external clock mode, <SIOS> must be cleared to "0" before next data shifting. If <SIOS> does not be cleared to "0" before next data shifting, SBI output dummy data and stopped.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	0	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	0	0	0	X	X	X	Starts transmission.

INTSBIx interrupt

SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.



(a) Internal clock

(b)External clock

Figure 13-16 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

		7	6	5	4	3	2	1	0	
if SBIxSR<SIOF> ≠ 0										Recognizes the completion of the transmission.
Then										
if SCK ≠ 1										Recognizes "1" is set to the SBxSCK pin by monitoring the port.
Then										
SBIxCR1	←	0	0	0	0	0	1	1	1	Completes the transmission by setting <SIOS> = 0.

(2) 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBIxCR1<SIOS> enables reception.Data is taken into the shift register from the SBxSI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIxDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIxDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether reception has come to an end.<SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1", the reception is aborted immediately and <SIOF> is cleared to "0". (The received data becomes invalid, and there is no need to read it out.)

Note: The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBIxCR1	←	1	0	1	1	0	X	X	X	Starts reception.

INTSBIx interrupt

Reg.	←	SBIxDBR	Reads the received data.

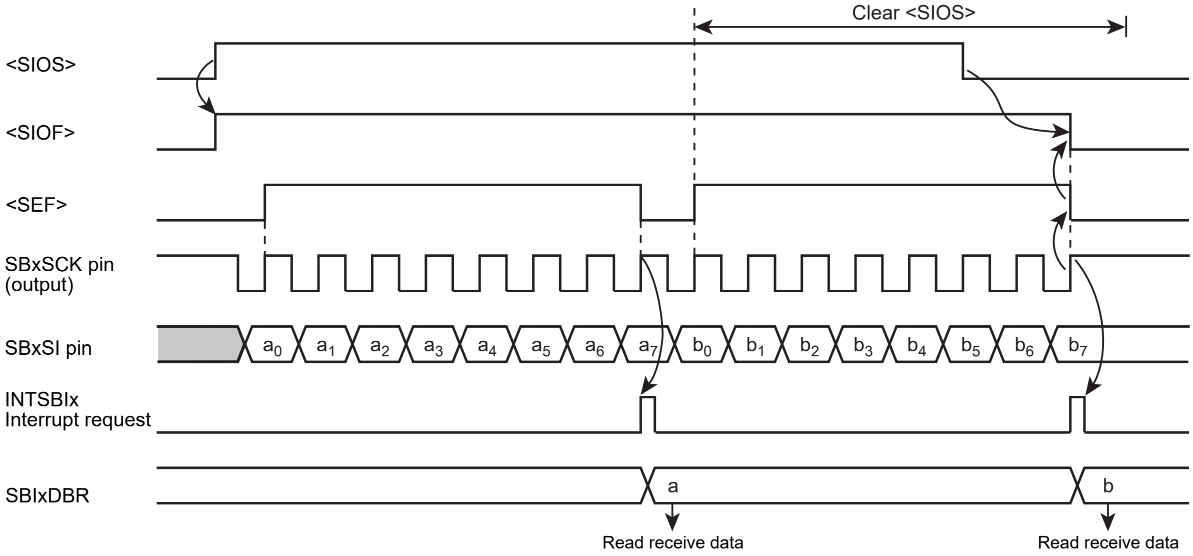


Figure 13-17 Receive Mode (Example: Internal Clock)

(3) 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIxDBR and setting SBIxCR1<SIOS> to "1" enables transmission and reception. The transmit data is output through the SBxSO pin at the falling of the serial clock, and the received data is taken in through the SBxSI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIxDBR and the INTSBIx interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIxDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started. The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SBxSCK pin.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBIxCR1<SIOINH> to "1" in the INTSBIx interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIxDBR. The program checks SBIxSR<SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception. If <SIOINH> is set to "1", the transmission and reception is aborted immediately and <SIOF> is cleared to "0".

Note: The contents of SBIxDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

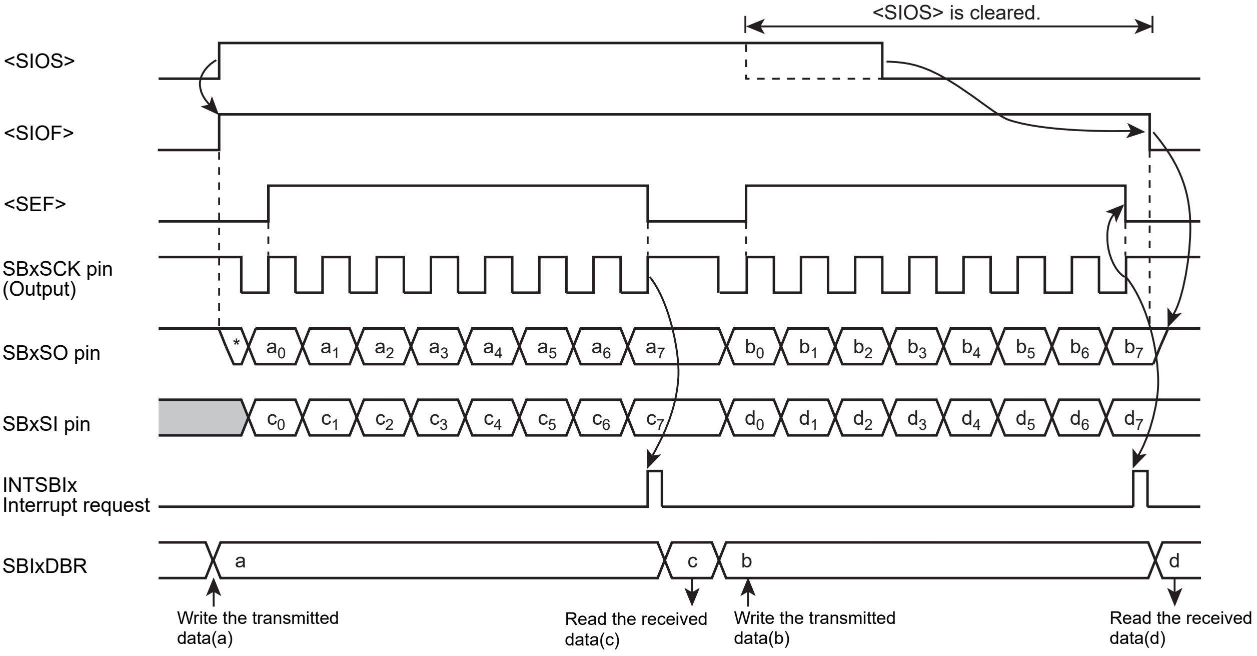


Figure 13-18 Transmit/Receive Mode (Example: Internal Clock)

		7	6	5	4	3	2	1	0	
SBIxCR1	←	0	1	1	0	0	X	X	X	Selects the transmit mode.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.
SBIxCR1	←	1	0	1	0	0	X	X	X	Starts reception/transmission.

INTSBIx interrupt

Reg.	←	SBIxDBR								Reads the received data.
SBIxDBR	←	X	X	X	X	X	X	X	X	Writes the transmit data.

(4) Data retention time of the last bit at the end of transmission

Under the condition SBIxCR1<SIOS>= "0", the last bit of the transmitted data retains the data of SBxSCK pin rising edge as shown below. Transmit mode and transmit/receive mode are the same.

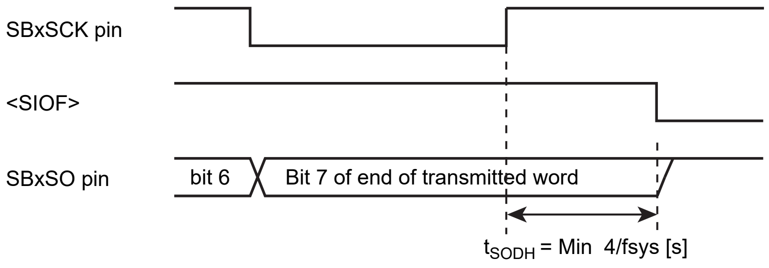


Figure 13-19 Data retention time of the last bit at the end of transmission

14. 12-Bit Analog-to-Digital Converter

14.1 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.

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.

14.2 Block Diagram

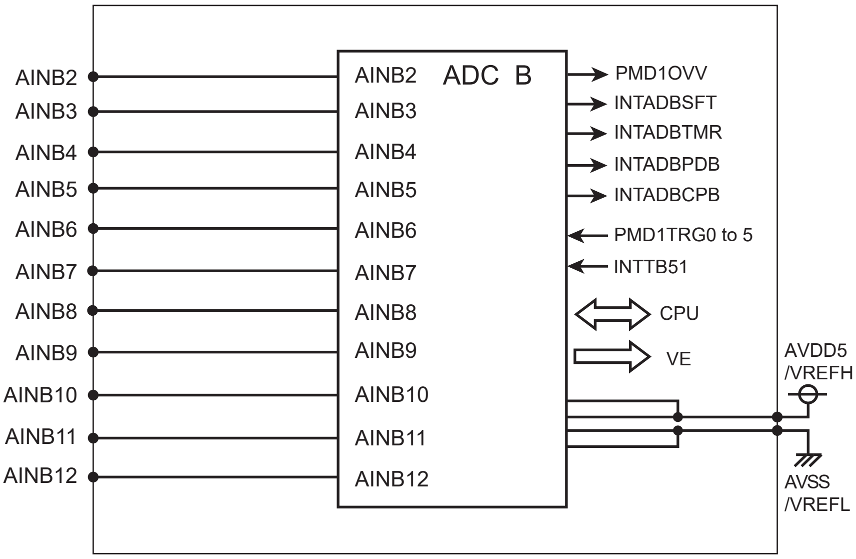


Figure 14-1 AD converters Block Diagram

14.3 List of Registers

The table below shows control registers and their addresses.

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

Register Name		Address(Base+)
Clock Setting Register	ADBCLK	0x0000
Mode Setting Register 0	ADBMOD0	0x0004
Mode Setting Register 1	ADBMOD1	0x0008
Mode Setting Register 2	ADBMOD2	0x000C
Monitoring Setting Register 0	ADBCMPCR0	0x0010
Monitoring Setting Register 1	ADBCMPCR1	0x0014
Conversion Result Compare Register 0	ADBCMP0	0x0018
Conversion Result Compare Register 1	ADBCMP1	0x001C
Conversion Result Register 0	ADBREG0	0x0020
Conversion Result Register 1	ADBREG1	0x0024
Conversion Result Register 2	ADBREG2	0x0028
Conversion Result Register 3	ADBREG3	0x002C
Conversion Result Register 4	ADBREG4	0x0030
Conversion Result Register 5	ADBREG5	0x0034
Conversion Result Register 6	ADBREG6	0x0038
Conversion Result Register 7	ADBREG7	0x003C
Conversion Result Register 8	ADBREG8	0x0040
Conversion Result Register 9	ADBREG9	0x0044
Conversion Result Register 10	ADBREG10	0x0048
Conversion Result Register 11	ADBREG11	0x004C
PMD Trigger Program Number Select Register 6	ADBPSEL6	0x0068
PMD Trigger Program Number Select Register 7	ADBPSEL7	0x006C
PMD Trigger Program Number Select Register 8	ADBPSEL8	0x0070
PMD Trigger Program Number Select Register 9	ADBPSEL9	0x0074
PMD Trigger Program Number Select Register 10	ADBPSEL10	0x0078
PMD Trigger Program Number Select Register 11	ADBPSEL11	0x007C
PMD Trigger Interrupt Select Register 0	ADBPINTS0	0x0080
PMD Trigger Interrupt Select Register 1	ADBPINTS1	0x0084
PMD Trigger Interrupt Select Register 2	ADBPINTS2	0x0088
PMD Trigger Interrupt Select Register 3	ADBPINTS3	0x008C
PMD Trigger Interrupt Select Register 4	ADBPINTS4	0x0090
PMD Trigger Interrupt Select Register 5	ADBPINTS5	0x0094
PMD Trigger Program Register 0	ADBPSET0	0x0098
PMD Trigger Program Register 1	ADBPSET1	0x009C
PMD Trigger Program Register 2	ADBPSET2	0x00A0
PMD Trigger Program Register 3	ADBPSET3	0x00A4
PMD Trigger Program Register 4	ADBPSET4	0x00A8
PMD Trigger Program Register 5	ADBPSET5	0x00AC
Timer Trigger Program Registers 0 to 3	ADBTSET03	0x00B0
Timer Trigger Program Registers 4 to 7	ADBTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADBTSET811	0x00B8
Software Trigger Program Registers 0 to 3	ADBSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADBSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADBSSET811	0x00C4
Constant Conversion Program Registers0 to 3	ADBASET03	0x00C8

Register Name		Address(Base+)
Constant Conversion Program Registers 4 to 7	ADBASET47	0x00CC
Constant Conversion Program Registers 8 to 11	ADBASET811	0x00D0
Mode Setting Register 3	ADBMOD3	0x00D4

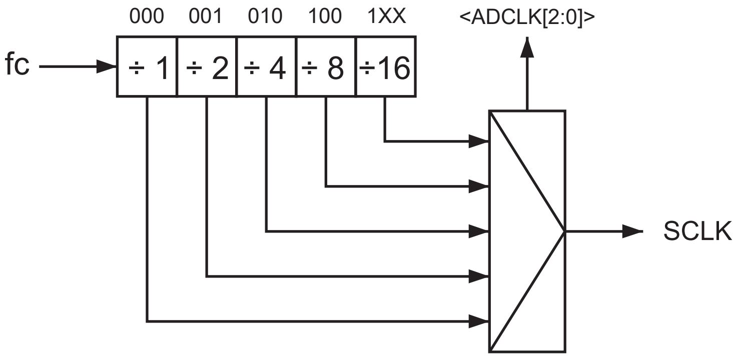
14.4 Register Descriptions

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

14.4.1 ADBCLK (Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-3	TSH[3:0]	R/W	Write as "1001".
2-0	ADCLK[2:0]	R/W	AD prescaler output (SCLK) select 000: fc (Note1) 001: fc/2 010: fc/4 011: fc/8 1xx: fc/16



Note 1: Frequency of SCLK can be used up to 40MHz.

Note 2: AD conversion is performed at the clock frequency selected in this register. The conversion clock frequency must be selected to ensure the guaranteed accuracy.

Note 3: The conversion clock must not be changed while AD conversion is in progress.

14.4.2 ADBMOD0 (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.

14.4.3 ADBMOD1 (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

14.4.4 ADBMOD2 (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".

14.4.5 ADBMOD3 (Mode Setting Register 3)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-11	−	R/W	Write as "0".
10	−	R/W	Write as "1".
9	−	R/W	Write as "0".
8	RCUT	R/W	ADC operational control 2 0: Enable 1: Disable When ADC is operated, write to "0" in advance. While ADC stops the operation, it can be reduced a power consumption by setting to "1".
7	−	R/W	Write as "0".
6	−	R/W	Write as "1".
5-3	PMODE[2:0]	R/W	Write as "100".
2-0	−	R/W	Write as "0".

Note: ADBMOD3<PMODE[2:0]> must be set to "100". And do not change other bits in ADBMOD3 register.

14.4.6 ADBCMPCR0(Monitoring Setting Register 0)

After fixing the conversion result, the interrupt signal (INTADBCPn) 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 The ADBCMPCR0 and ADBCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
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: ADBREG0 / 0100: ADBREG4 / 1000: ADBREG8 0001: ADBREG1 / 0101: ADBREG5 / 1001: ADBREG9 0010: ADBREG2 / 0110: ADBREG6 / 1010: ADBREG10 0011: ADBREG3 / 0111: ADBREG7 / 1011: ADBREG11

14.4.7 ADBCMPCR1(Monitoring Setting Register 1)

After fixing the conversion result, the interrupt signal (INTADBCPn) 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 The ADBCMPCR0 and ADBCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.
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: ADBREG0 0001: ADBREG1 0010: ADBREG2 0011: ADBREG3 0100: ADBREG4 0101: ADBREG5 0110: ADBREG6 0111: ADBREG7 1000: ADBREG8 1001: ADBREG9 1010: ADBREG10 1011: ADBREG11

14.4.8 ADBCMP0(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".

14.4.9 ADBCMP1(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".

14.4.10 ADBREG0(Conversion Result Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR00							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR00				-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR00[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR0	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG0 is read and is cleared when the low-order byte of ADBREG0 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 ADBREG0 register and is cleared when the low-order byte of ADBREG0 is read.

14.4.11 ADBREG1(Conversion Result Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR10[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR1	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG1 is read and is cleared when the low-order byte of ADBREG1 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 ADBREG1 register and is cleared when the low-order byte of ADBREG1 is read.

14.4.12 ADBREG2(Conversion Result Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR20							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR20				-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR20[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR2	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG2 is read and is cleared when the low-order byte of ADBREG2 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 ADBREG2 register and is cleared when the low-order byte of ADBREG2 is read.

14.4.13 ADBREG3(Conversion Result Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR30							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR30				-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR30[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR3	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG3 is read and is cleared when the low-order byte of ADBREG3 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 ADBREG3 register and is cleared when the low-order byte of ADBREG3 is read.

14.4.14 ADBREG4(Conversion Result Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR40							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR40				-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR40[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR4	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG4 is read and is cleared when the low-order byte of ADBREG4 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 ADBREG4 register and is cleared when the low-order byte of ADBREG4 is read.

14.4.15 ADBREG5(Conversion Result Register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR50							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR50				-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR50[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR5	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG5 is read and is cleared when the low-order byte of ADBREG5 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 ADBREG5 register and is cleared when the low-order byte of ADBREG5 is read.

14.4.16 ADBREG6(Conversion Result Register 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR60							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR60				-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR60[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR6	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG6 is read and is cleared when the low-order byte of ADBREG6 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 ADBREG6 register and is cleared when the low-order byte of ADBREG6 is read.

14.4.17 ADBREG7(Conversion Result Register 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR70							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR70				-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR70[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR7	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG7 is read and is cleared when the low-order byte of ADBREG7 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 ADBREG7 register and is cleared when the low-order byte of ADBREG7 is read.

14.4.18 ADBREG8(Conversion Result Register 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR80							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR80				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR80[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR8	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG8 is read and is cleared when the low-order byte of ADBREG8 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 ADBREG8 register and is cleared when the low-order byte of ADBREG8 is read.

14.4.19 ADBREG9(Conversion Result Register 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR90							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR90				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR90[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR9	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG9 is read and is cleared when the low-order byte of ADBREG9 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 ADBREG9 register and is cleared when the low-order byte of ADBREG9 is read.

14.4.20 ADBREG10(Conversion Result Register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR100							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR100				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR100[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR10	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG10 is read and is cleared when the low-order byte of ADBREG10 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 ADBREG10 register and is cleared when the low-order byte of ADBREG10 is read.

14.4.21 ADBREG11(Conversion Result Register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR110							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR110				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR110[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR11	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADBREG11 is read and is cleared when the low-order byte of ADBREG11 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 ADBREG11 register and is cleared when the low-order byte of ADBREG11 is read.

14.4.22 PMD Trigger Program Registers

AD conversion can be started by a trigger from the PMD (programmable motor driver).

The PMD trigger program registers are used to specify the program to be started by each of six triggers generated by the PMD, to select the interrupt to be generated upon completion of the program and to select the AIN input to be used.

The PMD trigger program registers include three types of registers.

- PMD Trigger Program Number Select Register (ADBPSEL6 to ADBPSEL11)

  The PMD Trigger Program Number Select Register (ADBPSELn) specifies the program to be started by each of six AD conversion start signals corresponding to six triggers(PMD1TRG0 to 5) generated by the PMD. Programs 0 to 5 are available.

  "ADBPSEL6 to ADBPSEL11" corresponds to "PMD1TRG0 to 5".
- PMD Trigger Interrupt Select Register (ADBPINTS0 to ADBPINTS5)

  The PMD Trigger Interrupt Select Registers (ADBPINTS0 to ADBPINTS5) select the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

  ADBPINTS0 corresponds to program 0, and it exists to ADBPINT5 (program 5).
- PMD Trigger Program Register (ADBPSET0 to ADBPSET5)

  The PMD Trigger Program Setting Registers (ADBPSET0 to ADBPSET5) 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 ADBPSETn0 to ADBPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADBREG0 to ADBREG3).

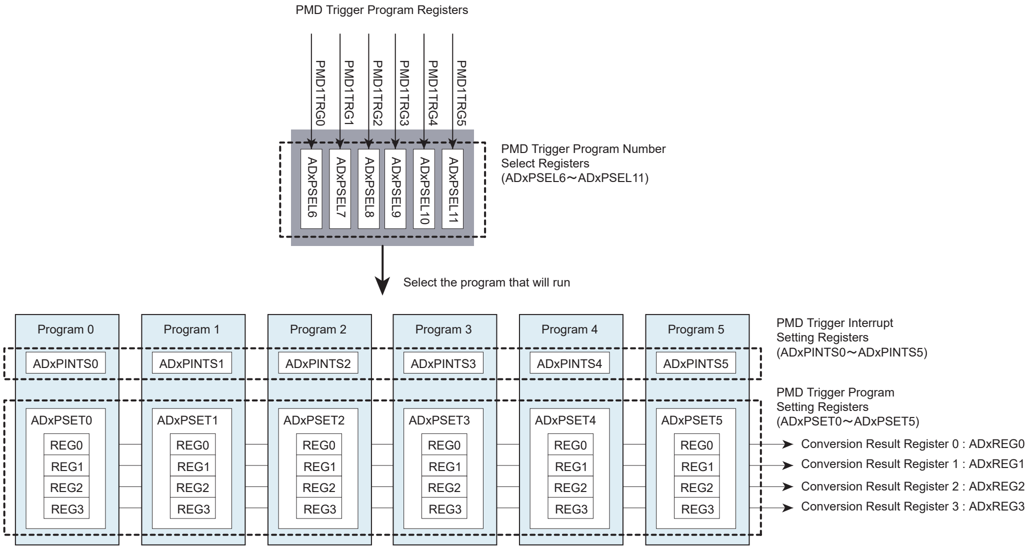

PMD Trigger Program Registers
PMD1TRG0
PMD1TRG1
PMD1TRG2
PMD1TRG3
PMD1TRG4
PMD1TRG5
ADxPSEL6
ADxPSEL7
ADxPSEL8
ADxPSEL9
ADxPSEL10
ADxPSEL11
PMD Trigger Program Number Select Registers (ADxPSEL6～ADxPSEL11)
Select the program that will run
Program 0
Program 1
Program 2
Program 3
Program 4
Program 5
ADxPINTS0
ADxPINTS1
ADxPINTS2
ADxPINTS3
ADxPINTS4
ADxPINTS5
PMD Trigger Interrupt Setting Registers (ADxPINTS0～ADxPINTS5)
ADxPSET0
ADxPSET1
ADxPSET2
ADxPSET3
ADxPSET4
ADxPSET5
REG0
REG1
REG2
REG3
PMD Trigger Program Setting Registers (ADxPSET0～ADxPSET5)
Conversion Result Register 0 : ADxREG0
Conversion Result Register 1 : ADxREG1
Conversion Result Register 2 : ADxREG2
Conversion Result Register 3 : ADxREG3

Figure 14-2 PMD Trigger Program Registers

14.4.22.1 ADBPSEL6 to ADBPSEL11(PMD Trigger Program Number Select Register 6 to 11)

ADBPSEL6: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 14-1)

ADBPSEL7: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 14-1)

ADBPSEL8: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 14-1)

ADBPSEL9: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 14-1)

ADBPSEL10: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 14-1)

ADBPSEL11: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 14-1)

Table 14-1 Program number select

<PMDS6[2:0]> to <PMDS11[2:0]>	
000	Program0
001	Program1
010	Program2
011	Program3
100	Program4
101	Program5
110	reserved
111	reserved

14.4.22.2 ADBPINTS0 to 5(PMD Trigger Interrupt Select Register 0 to 5)

ADBPINTS0:PMD Trigger Interrupt Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL0	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL0[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADBPDB 11: No interrupt output The starting interrupt is selected for program 0.

ADBPINTS1:PMD Trigger Interrupt Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL1	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL1[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADBPDB 11: No interrupt output The starting interrupt is selected for program 1.

ADBPINTS2:PMD Trigger Interrupt Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL2	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL2[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADBPDB 11: No interrupt output The starting interrupt is selected for program 2.

ADBPINTS3:PMD Trigger Interrupt Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL3	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL3[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADBPDB 11: No interrupt output The starting interrupt is selected for program 3.

ADBPINTS4:PMD Trigger Interrupt Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL4	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL4[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADBPDB 11: No interrupt output The starting interrupt is selected for program 4.

ADBPINTS5:PMD Trigger Interrupt Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL5	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL5[1:0]	R/W	Interrupt select 00:No interrupt output 01:Reserved 10:INTADBPDB 11: No interrupt output The starting interrupt is selected for program 5.

14.4.22.3 ADBPSET0 to 5(PMD Trigger Program Register 0 to 5)

Each ADBPSETn (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)

Setting the <ENSPnm> to "1" enables the the <UVWISnm[1:0]>, the <AINSPnm[4:0]> bits are used to select the AIN pin to be used. With these conditions, the AD conversion is started and then stored into a conversion result register.

ADBREGm ADBPSETn	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 14-2 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:AINB2
0_0011	:AINB3
0_0100	:AINB4
0_0101	:AINB5
0_0110	:AINB6
0_0111	:AINB7
0_1000	:AINB8
0_1001	:AINB9
0_1010	:AINA10
0_1011	:AINB11
0_1100	:AINB12
0_1101 to 1_1111	:Reserved

ADBPSET0: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	ADBREG3 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 14-2 Select the AIN pin".
23	ENSP02	R/W	ADBREG2 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 14-2 Select the AIN pin".
15	ENSP01	R/W	ADBREG1 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 14-2 Select the AIN pin".
7	ENSP00	R/W	ADBREG0 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 14-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADBPSET1: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	ADBREG3 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 14-2 Select the AIN pin".
23	ENSP12	R/W	ADBREG2 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 14-2 Select the AIN pin".
15	ENSP11	R/W	ADBREG1 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 14-2 Select the AIN pin".
7	ENSP10	R/W	ADBREG0 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 14-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADBPSET2: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	ADBREG3 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 14-2 Select the AIN pin".
23	ENSP22	R/W	ADBREG2 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 14-2 Select the AIN pin".
15	ENSP21	R/W	ADBREG1 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 14-2 Select the AIN pin".
7	ENSP20	R/W	ADBREG0 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 14-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADBPSET3: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	ADBREG3 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 14-2 Select the AIN pin".
23	ENSP32	R/W	ADBREG2 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 14-2 Select the AIN pin".
15	ENSP31	R/W	ADBREG1 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 14-2 Select the AIN pin".
7	ENSP30	R/W	ADBREG0 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 14-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADBPSET4: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	ADBREG3 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 14-2 Select the AIN pin".
23	ENSP42	R/W	ADBREG2 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 14-2 Select the AIN pin".
15	ENSP41	R/W	ADBREG1 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 14-2 Select the AIN pin".
7	ENSP40	R/W	ADBREG0 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 14-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADBPSET5: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	ADBREG3 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 14-2 Select the AIN pin".
23	ENSP52	R/W	ADBREG2 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 14-2 Select the AIN pin".
15	ENSP51	R/W	ADBREG1 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 14-2 Select the AIN pin".
7	ENSP50	R/W	ADBREG0 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 14-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

14.4.23 ADBTSET03 / ADBTSET47 / ADBTSET811 (Timer Trigger Program Registers)

AD conversion can be started by INTTB51 generated from Timer5(TMRB5) as a trigger. There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to "1" enables the ADBTSETm register. The <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 : INTADBTMR is generated.

(m=0 to 11)

Table 14-3 Select the AIN pin

<AINST0 [4:0]> to <AINST11 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:AINB2
0_0011	:AINB3
0_0100	:AINB4
0_0101	:AINB5
0_0110	:AINB6
0_0111	:AINB7
0_1000	:AINB8
0_1001	:AINB9
0_1010	:AINA10
0_1011	:AINB11
0_1100	:AINB12
0_1101 to 1_1111	:Reserved

ADBTSET03: 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	ADBREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST3[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
23	ENST2	R/W	ADBREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST2[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
15	ENST1	R/W	ADBREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST1[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
7	ENST0	R/W	ADBREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST0[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".

ADBTSET47: 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	ADBREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST7[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
23	ENST6	R/W	ADBREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST6[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
15	ENST5	R/W	ADBREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST5[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
7	ENST4	R/W	ADBREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST4[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".

ADBTSET811: 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	ADBREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST11[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
23	ENST10	R/W	ADBREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST10[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
15	ENST9	R/W	ADBREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST9[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".
7	ENST8	R/W	ADBREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST8[4:0]	R/W	AIN select Refer to "Table 14-3 Select the AIN pin".

14.4.24 ADBSSET03 / ADBSSET47 / ADBSSET811(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 ADBSSETm 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 :INTADBSFT is generated.

(m=0 to 11)

Table 14-4 Select the AIN pin

<AINSS0 [4:0]> to <AINSS11 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:AINB2
0_0011	:AINB3
0_0100	:AINB4
0_0101	:AINB5
0_0110	:AINB6
0_0111	:AINB7
0_1000	:AINB8
0_1001	:AINB9
0_1010	:AINA10
0_1011	:AINB11
0_1100	:AINB12
0_1101 to 1_1111	:Reserved

ADBSSET03: 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	ADBREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS3[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
23	ENSS2	R/W	ADBREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS2[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
15	ENSS1	R/W	ADBREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS1[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
7	ENSS0	R/W	ADBREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS0[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".

ADBSSET47: 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	ADBREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS7[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
23	ENSS6	R/W	ADBREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS6[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
15	ENSS5	R/W	ADBREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS5[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
7	ENSS4	R/W	ADBREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS4[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".

ADBSSET811: 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	ADBREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS11[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
23	ENSS10	R/W	ADBREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS10[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
15	ENSS9	R/W	ADBREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS9[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".
7	ENSS8	R/W	ADBREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS8[4:0]	R/W	AIN select Refer to "Table 14-4 Select the AIN pin".

14.4.25 ADBASET03 / ADBASET47 / ADBASET811(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 ADBASETm register. The <AINSAm[4:0] > are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m=0 to 11)

Table 14-5 Select the AIN pin

<AINSA0 [4:0]> to <AINSA11 [4:0]>	ADC Unit B
0_0000	:Reserved
0_0001	:Reserved
0_0010	:AINB2
0_0011	:AINB3
0_0100	:AINB4
0_0101	:AINB5
0_0110	:AINB6
0_0111	:AINB7
0_1000	:AINB8
0_1001	:AINB9
0_1010	:AINA10
0_1011	:AINB11
0_1100	:AINB12
0_1101 to 1_1111	:Reserved

ADBASET03: 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	ADBREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA3[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
23	ENSA2	R/W	ADBREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA2[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
15	ENSA1	R/W	ADBREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA1[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
7	ENSA0	R/W	ADBREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA0[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".

ADBASET47: 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	ADBREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA7[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
23	ENSA6	R/W	ADBREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA6[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
15	ENSA5	R/W	ADBREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA5[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
7	ENSA4	R/W	ADBREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA4[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".

ADBASET811: Constant 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	ADBREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA11[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
23	ENSA10	R/W	ADBREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA10[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
15	ENSA9	R/W	ADBREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA9[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".
7	ENSA8	R/W	ADBREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA8[4:0]	R/W	AIN select Refer to "Table 14-5 Select the AIN pin".

14.5 Operation Descriptions

14.5.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the VREFHB and VREFLB pins are used in ADC B. When ADBMOD3<RCUT> is set to "1", VREFHB-VREFLB is switched off from the condition switched on.

14.5.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD trigger (See "14.4.22 PMD Trigger Program Registers")
- Timer trigger (TMRB5) (See "14.4.23 Timer Trigger Program Registers.")
- Software trigger (See "14.4.24 Software Trigger Program Registers.")

These start triggers are given priorities as shown below.

PMD trigger 0 > ···· > PMD trigger 5 > Timer trigger > Software trigger > constant trigger

When a higher-priority trigger occurs while an AD conversion is in progress, a higher-priority trigger is handled stop the ongoing program and start AD conversion correspond to a higher-priority trigger number.

When the PMD trigger occurs while a PMD triggered AD conversion is in progress, the PMD trigger is handled after the ongoing program is completed.

It has some delay from generation of trigger to start of AD conversion. The delay depends on the trigger. The following timing chart and table show the delay.

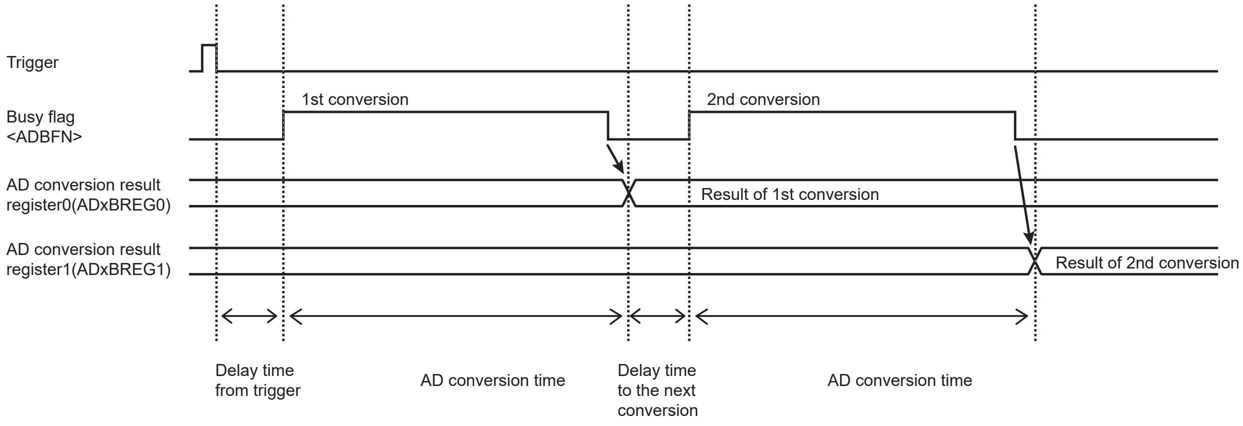


Figure 14-3 Timing chart of AD conversion

Table 14-6 AD conversion time and delay 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]	–	2.00		2.00	
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.

14.5.3 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 ADBCMPCR0<CMP0EN> or ADBCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by ADBCMCR0<ADBIG0>/ADBCMPR1<ADBIG1>, the interrupt (INTADBCPA for ADBCMPCR0, INTADBCPB for ADBCMPCR1) is generated. The comparison is executed at the timing of storing the conversion result into the register.

Note 1: The AD conversion result store flag (<ADRxRF>) is not cleared by the comparison function.

Note 2: The comparison function differs from reading the conversion result by software. Therefore, if the next conversion is completed without reading the previous result, the overrun flag (<OVRx>) is set.

14.6 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

14.6.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADBSSET03, ADBSSET47 and ADBSSET811.(Figure 14-4)

If the ADBMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.(Figure 14-5)

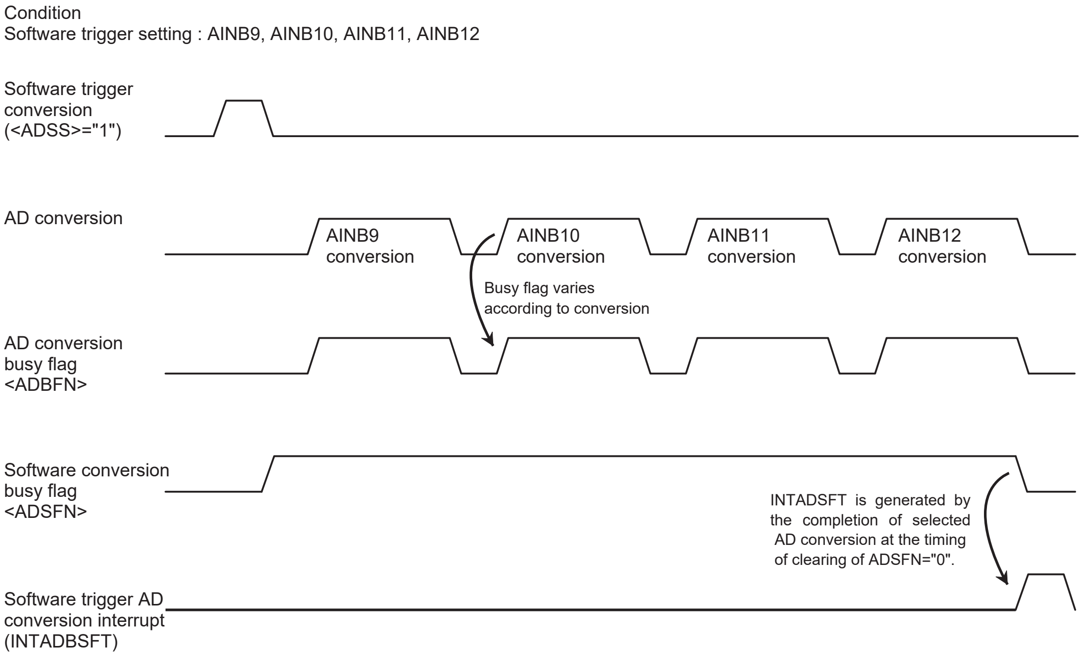

Condition
Software trigger setting : AINB9, AINB10, AINB11, AINB12
Software trigger conversion (<ADSS>="1")
AD conversion
AINB9 conversion
AINB10 conversion
AINB11 conversion
AINB12 conversion
Busy flag varies according to conversion
AD conversion busy flag <ADBFN>
Software conversion busy flag <ADSFN>
INTADSFT is generated by the completion of selected AD conversion at the timing of clearing of ADSFN="0".
Software trigger AD conversion interrupt (INTADBSFT)

Figure 14-4 Software trigger AD conversion

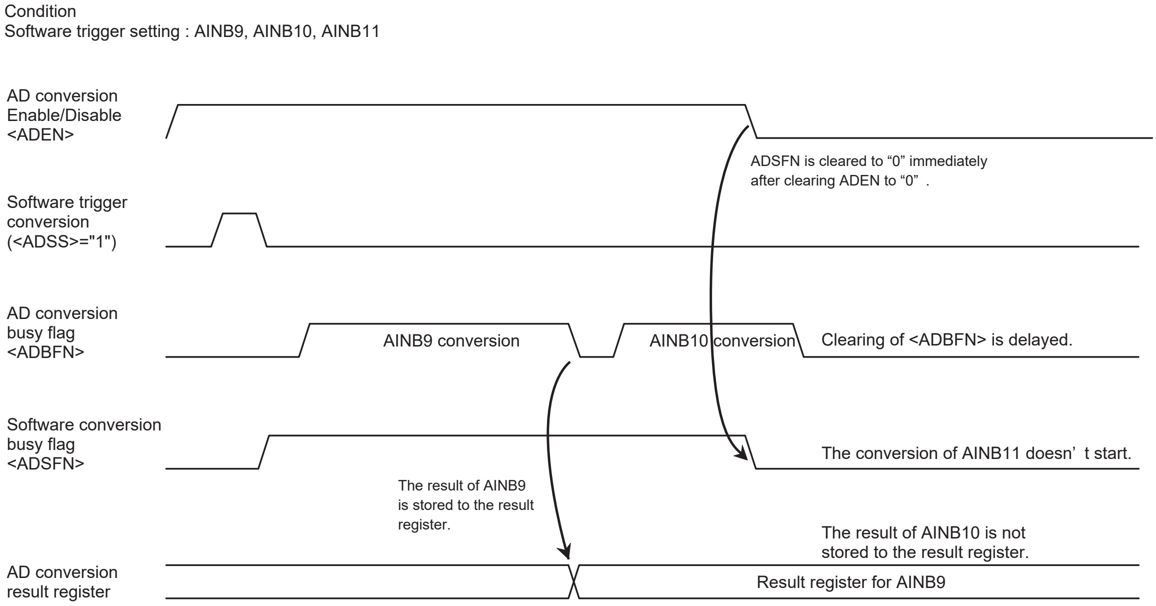

Condition
Software trigger setting : AINB9, AINB10, AINB11
AD conversion Enable/Disable <ADEN>
ADSFN is cleared to “0” immediately after clearing ADEN to “0” .
Software trigger conversion (<ADSS>="1")
AD conversion busy flag <ADBFN>
AINB9 conversion
AINB10 conversion
Clearing of <ADBFN> is delayed.
Software conversion busy flag <ADSFN>
The conversion of AINB11 doesn’ t start.
The result of AINB9 is stored to the result register.
The result of AINB10 is not stored to the result register.
AD conversion result register
Result register for AINB9

Figure 14-5 Writing "0" to <ADEN> during the software trigger AD conversion

14.6.2 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 14-6)

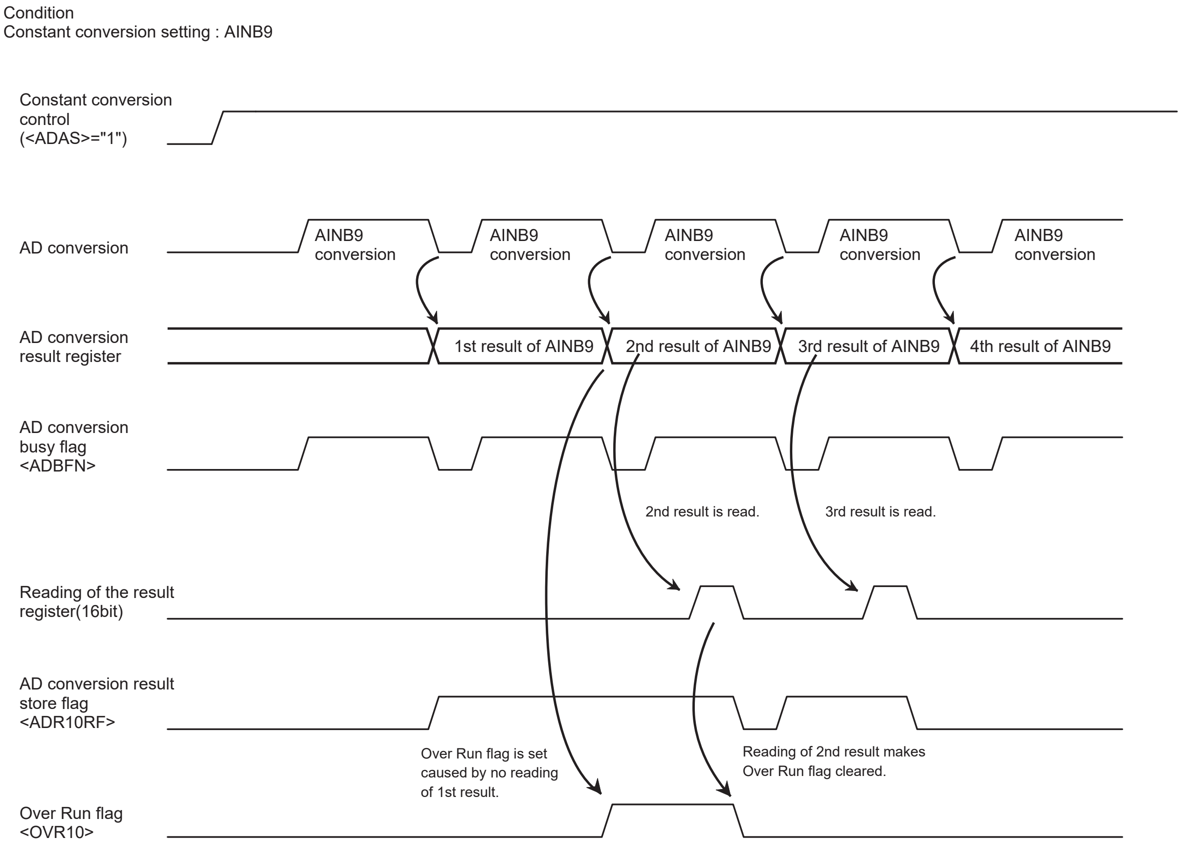

Condition
Constant conversion setting : AINB9
Constant conversion control (<ADAS>="1")
AD conversion
AINB9 conversion
AINB9 conversion
AINB9 conversion
AINB9 conversion
AINB9 conversion
AD conversion result register
1st result of AINB9
2nd result of AINB9
3rd result of AINB9
4th result of AINB9
AD conversion busy flag <ADBFN>
2nd result is read.
3rd result is read.
Reading of the result register(16bit)
AD conversion result store flag <ADR10RF>
Over Run flag is set caused by no reading of 1st result.
Reading of 2nd result makes Over Run flag cleared.
Over Run flag <OVR10>

Figure 14-6 Constant conversion

14.6.3 AD conversion by trigger

When the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately and start AD conversion correspond to PMD trigger. (Figure 14-7) After the completion of conversion by PMD trigger, the software trigger conversion starts from the beginning programmed setting. When the timer trigger is occurred, also same response. (Figure 14-8)

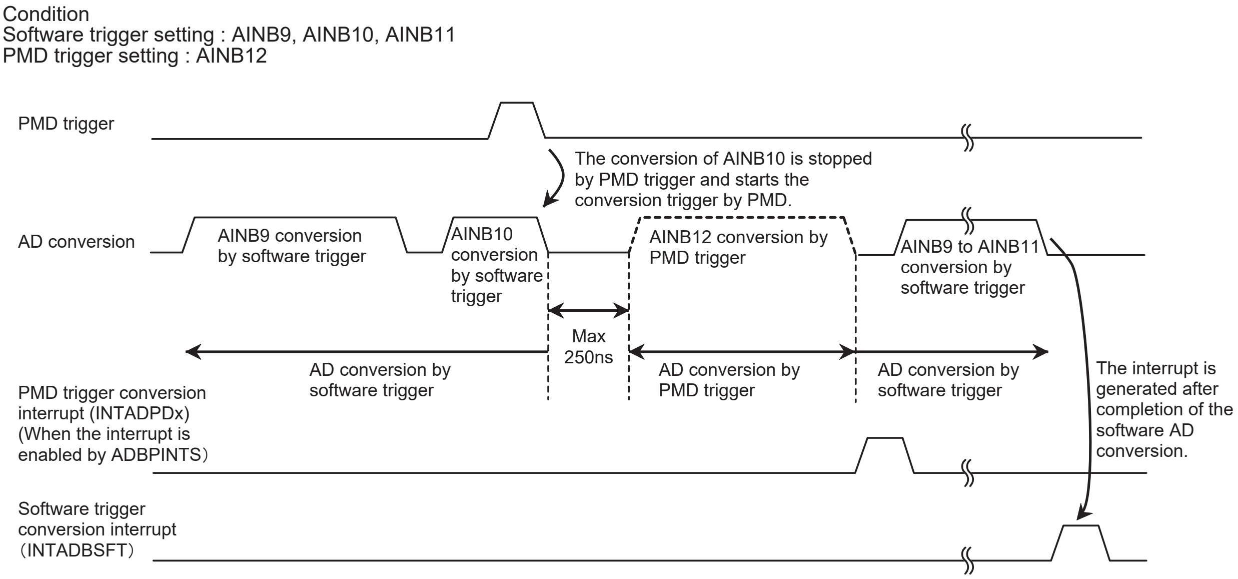

Condition
Software trigger setting : AINB9, AINB10, AINB11
PMD trigger setting : AINB12
PMD trigger
The conversion of AINB10 is stopped by PMD trigger and starts the conversion trigger by PMD.
AD conversion
AINB9 conversion by software trigger
AINB10 conversion by software trigger
AINB12 conversion by PMD trigger
AINB9 to AINB11 conversion by software trigger
Max 250ns
AD conversion by software trigger
AD conversion by PMD trigger
AD conversion by software trigger
The interrupt is generated after completion of the software AD conversion.
PMD trigger conversion interrupt (INTADPDx) (When the interrupt is enabled by ADBPINTS)
Software trigger conversion interrupt （INTADBSFT）

Figure 14-7 AD conversion by PMD trigger

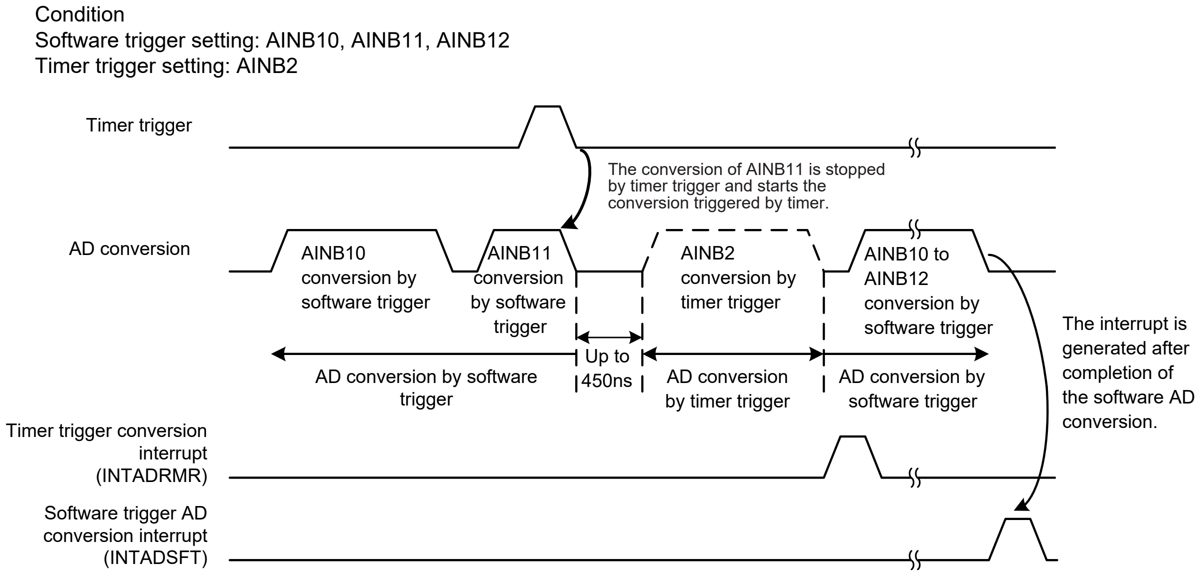

Condition
Software trigger setting: AINB10, AINB11, AINB12
Timer trigger setting: AINB2
Timer trigger
The conversion of AINB11 is stopped by timer trigger and starts the conversion triggered by timer.
AD conversion
AINB10 conversion by software trigger
AINB11 conversion by software trigger
AINB2 conversion by timer trigger
AINB10 to AINB12 conversion by software trigger
Up to 450ns
AD conversion by software trigger
AD conversion by timer trigger
AD conversion by software trigger
The interrupt is generated after completion of the software AD conversion.
Timer trigger conversion interrupt (INTADRMR)
Software trigger AD conversion interrupt (INTADSFT)

Figure 14-8 AD conversion by timer trigger

Note: When timer trigger is not used, do not use INTTB51. Set TB5IM<TBIM1> to "1".

14.7 Precautions on Use of AD Converter

During AD conversion, do not change the output data of port J/K/N, to avoid the influence on the conversion result.

The AD conversion result may vary by power supply fluctuation or environmental noise.

If an input or output flowing to the pin sharing with an AD input is changed while AD conversion is ongoing, or output current flowing to other pins that are specified as an output port fluctuates, precision of AD conversion may be lower.

Take measures against these problems by averaging of the multiple conversion results by the program.

15. Motor Control Circuit (PMD: Programmable Motor Driver)

The PMD of this product can control a three-phase motor such as vector motors in conjunction with a Vector Engine (VE+) and an analog/digital converter (ADC). Pulse-width modulation circuits, conduction control and synchronous trigger generators can be activated by commands from the Vector Engine. The synchronous trigger generation circuit can command the AD converter to start ADC conversion.

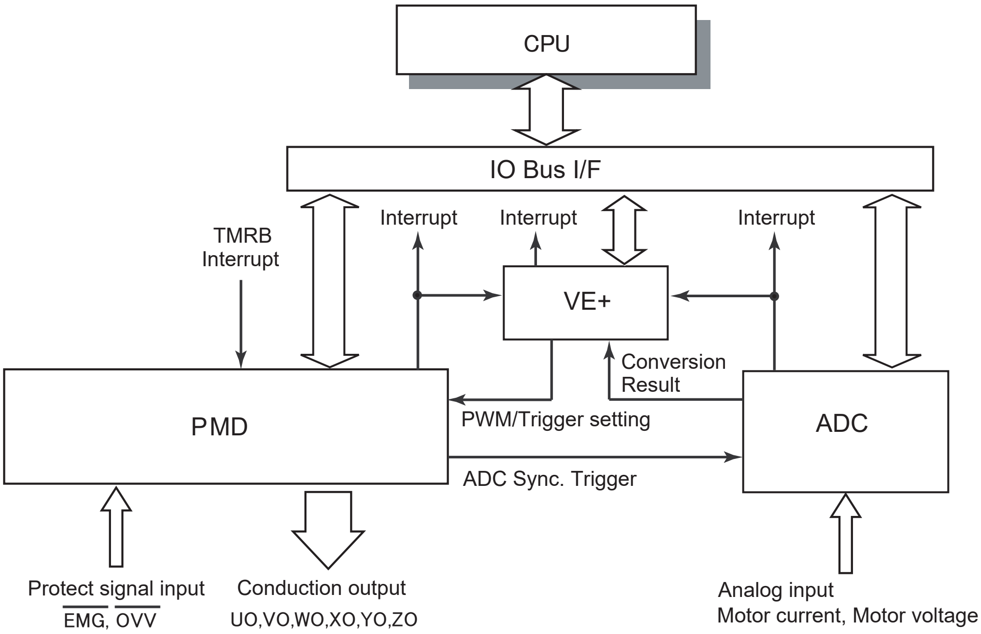


Figure 15-1 Block Diagram of Functions related to Motor Control

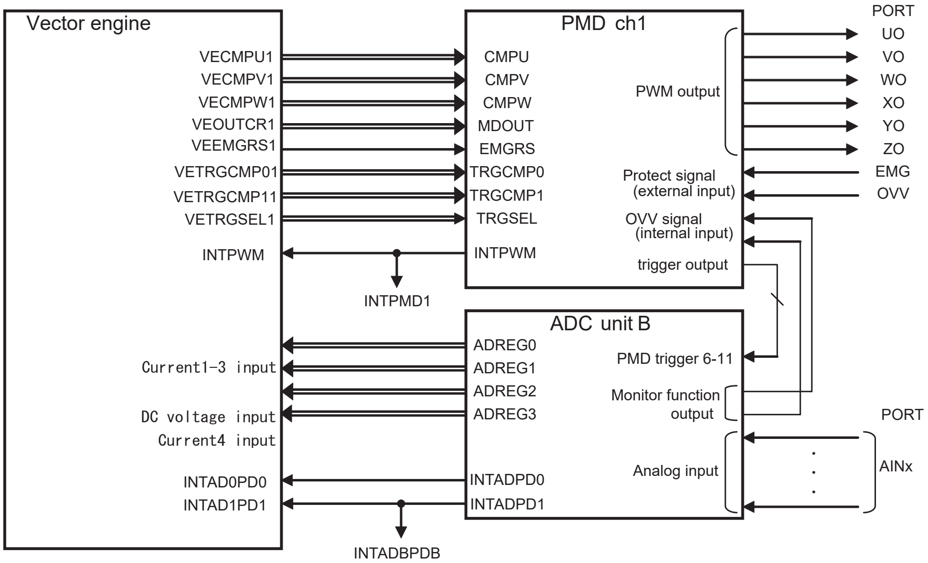


Figure 15-2 Related diagram of Motor control circuit, Vector engine and AD converter

15.1 PMD Circuit configuration

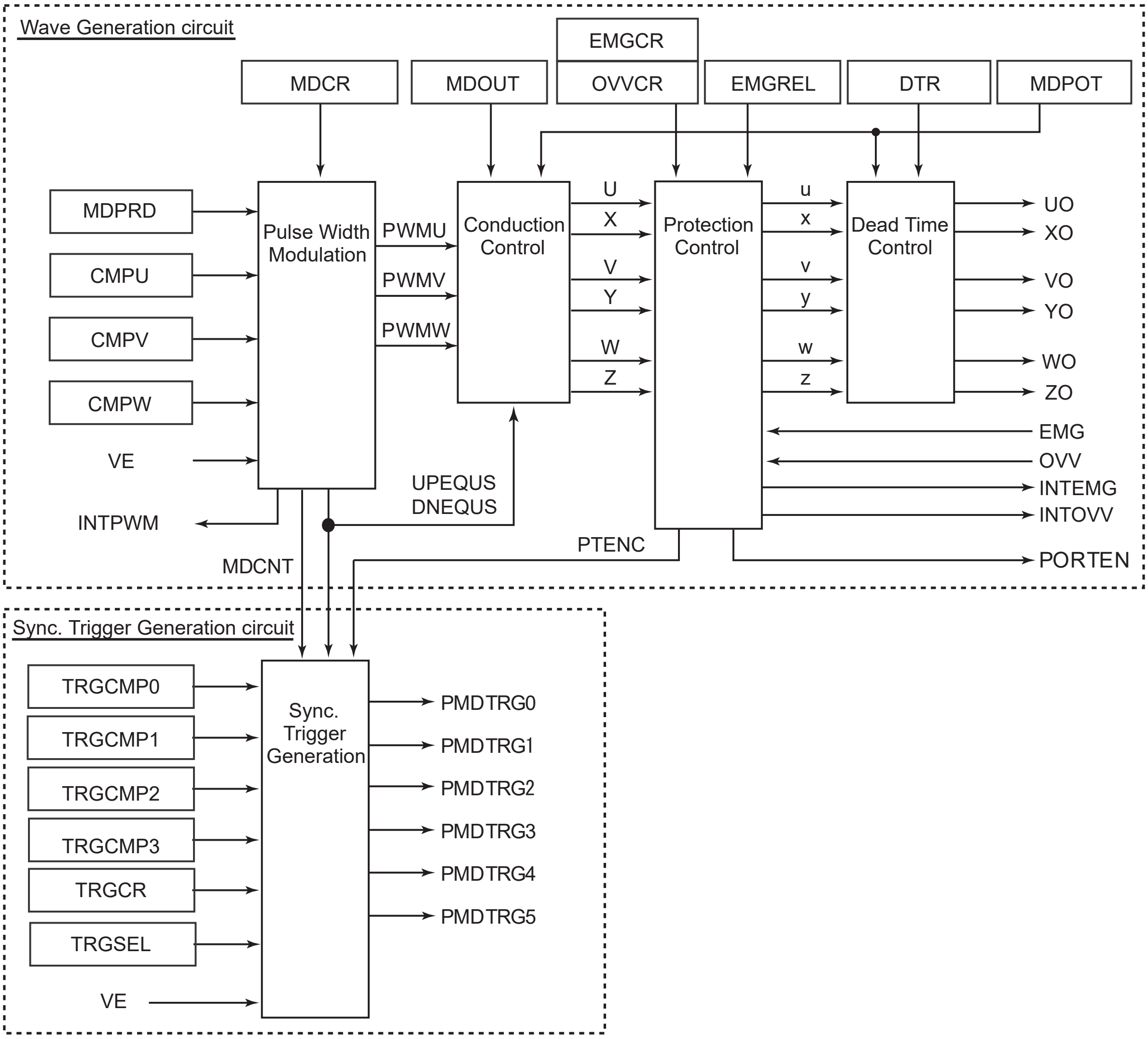


Figure 15-3 Block diagram of PMD Circuit

The PMD circuit consists of two blocks of a wave generation circuit and a sync trigger generation circuit. The wave generation circuit includes a pulse width modulation circuit, a conduction control circuit, a protection control circuit, a dead time control circuit.

- The pulse width modulation circuit has the common PWM carrier waveform and generates independent 3-phase PWM waveforms.
- The conduction control circuit determines the output pattern for each of the upper and lower sides of the U, V and W phases.
- The protection control circuit controls emergency output stop by EMG input and OVV input.
- The dead time control circuit prevents a short circuit which may occur when the upper side and lower side are switched.
- The sync trigger generation circuit generates sync trigger signals to the AD converter.

15.2 PMD Registers

The following table lists the control registers and their addresses:

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

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 Conduction 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
Trigger Update Timing Setting Register	PMDxTRGSYNCR	0x0064

15.2.1 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 Note: When the port is set to a function output (PWM output), the port disables output (high impedance) by setting <PWMEN> = "0". Note: Before enabling the PMD, Setting <PWMEN>="1"(enable) other relevant settings, such as output port polarity.

15.2.2 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 when a tool break occurs 00: Upper phases = High-z / lower phases = High-z 01: Upper phases = High-z / lower phases = PMD output 10: Upper phases = PMD output / lower phases = High-z 11: Upper phases = PMD output / lower phases = PMD output Sets the port output for both upper phase (UO/VO/WO) and the lower phase (XO/YO/ZO) when a tool break occurs in the use of ports for function output (PWM output). When a tool break occurs while "High-Z" is selected, the ports are disabled to output (high impedance). In other cases, external port outputs depend on PMD outputs.

Note 1: When <PWMEN>=0, output ports are disabled to output (high impedance) regardless of the PORTMD setting.

Note 2: When an EMG input occurs, port outputs are controlled depending by setting the PMDxEMGCR<EMGMD[1:0]>.

15.2.3 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	DCMPEN	-	-	-	MDSEL3	MDSEL2	MDSEL1	MDSEL0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7	DCMPEN	R/W	Automatic switching between VE register and PMD register 0: Disable switching control between 2 registers (using the register to set <MDSEL0> only). 1: Enable switching control between 2 registers (in up- and down-count state of the PWM counter). Note: Valid when <MDSEL0> ="1". Note: Valid when triangle carrier wave is selected (PMDxMDCR<PWMMD>="1").
6-4	−	R	Read as 0.
3	MDSEL3	R/W	Mode Select 3 0: Bus mode (using PMD register: PMDxTRGSEL) 1: VE mode (using VE register: VETRGSEL1)
2	MDSEL2	R/W	Mode Select 2 0: Bus mode (using PMD registers: PMDxTRGCMP0 and PMDxTRGCMP1) 1: VE mode (using VE registers: VETRGCMP01 and VETRGCMP11)
1	MDSEL1	R/W	Mode Select 1 0: Bus mode (using PMD register: PMDxMDOUT) 1: VE mode (using VE register: VEOUTCR1)
0	MDSEL0	R/W	Mode Select 0 0: Bus mode (using PMD registers: PMDxCMPU,PMDxCMPV1 and PMDxCMPW) 1: VE mode (using VE registers: VECMPU1,VECMPV1 and VECMPW1 registers and sets VEEMGRS1 register enabled)

15.2.4 Pulse Width Modulation Circuit

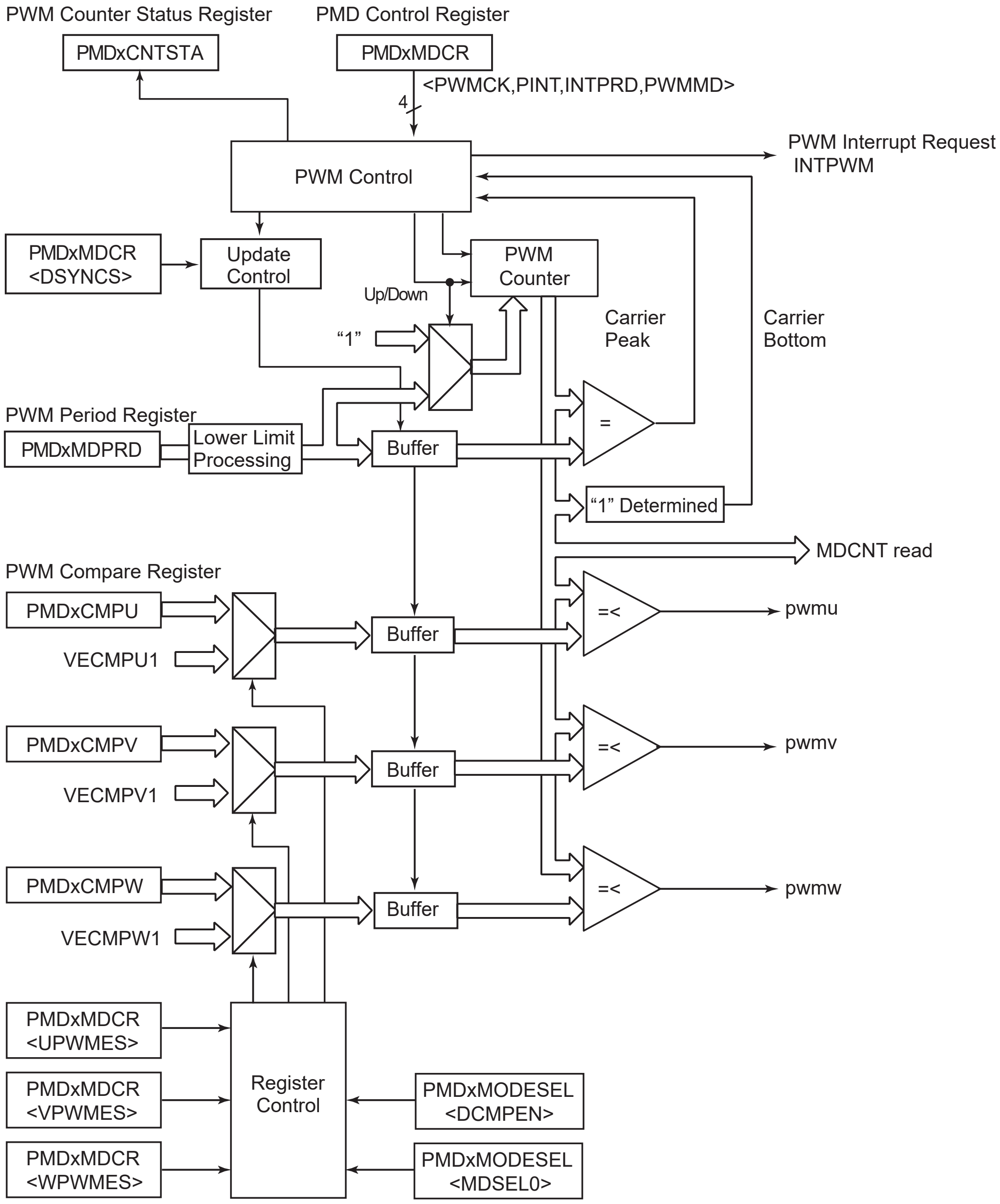


Figure 15-4 Pulse Width Modulation Circuit

The pulse width modulation circuit has a 16-bit PWM up-/down-counter and generates PWM carrier waveforms with a resolution of 1/fsys (25[ns] at 40[MHz]). 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 4/fsys (100[ns] at 40[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). (Refer to "Figure 15-5 PWM Waveforms".) In the triangular wave mode, PWM waveform can be selected from the center PWM, the fixed falling edge PWM and the fixed rising edge PWM. (Refer to "Figure 15-6 Waveforms of PWM triangular wave carrier using fixed edge".)

1. Setting the PWM period

   The PWM period is determined by the PMDxMDPRD register. This register is double-buffered. The subsequent stage buffer is updated at every PWM period. It is also possible to update at every half PWM period. (Refer to "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing".)

$$\text{Sawtooth wave PWM : PMDxMDPRD register Value} = \frac{\text{Oscillation frequency[Hz]}}{\text{PWM frequency[Hz]}}$$

$$\text{Triangular wave PWM : PMDxMDPRD register value} = \frac{\text{Oscillation frequency[Hz]}}{\text{PWM frequency[Hz] x 2}}$$

2. Compare function

   The pulse width modulation circuit generates PWM waveforms of the desired duty by comparing the magnitude of the PWM compare registers (PMDxCMPU/V/W) and the PWM carrier which is generated by the PWM counter (PMDxMDCNT <MDCNT[15:0]>).

   The PWM compare register of each phase has a double-buffered register. The PWM compare register value is loaded into the subsequent stage buffer at every PWM period. It is also possible to update at every half a PWM period. (Refer to "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing".)

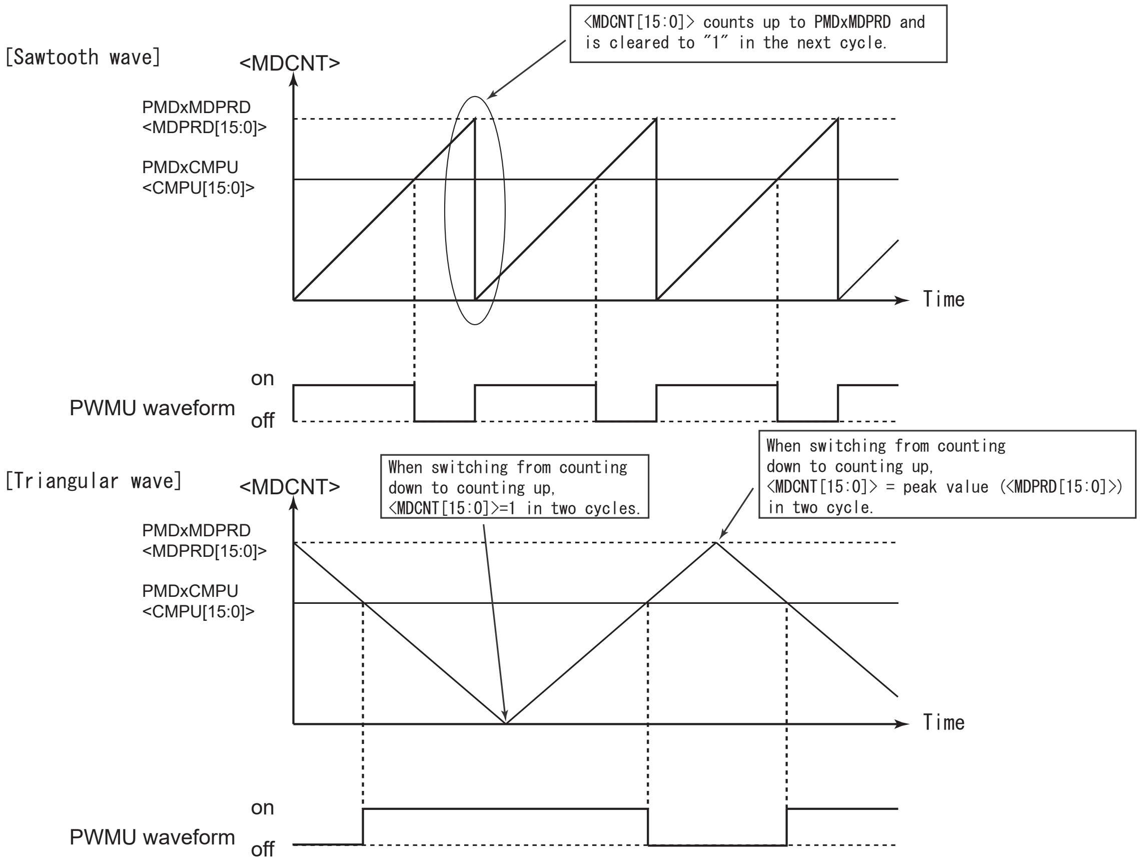
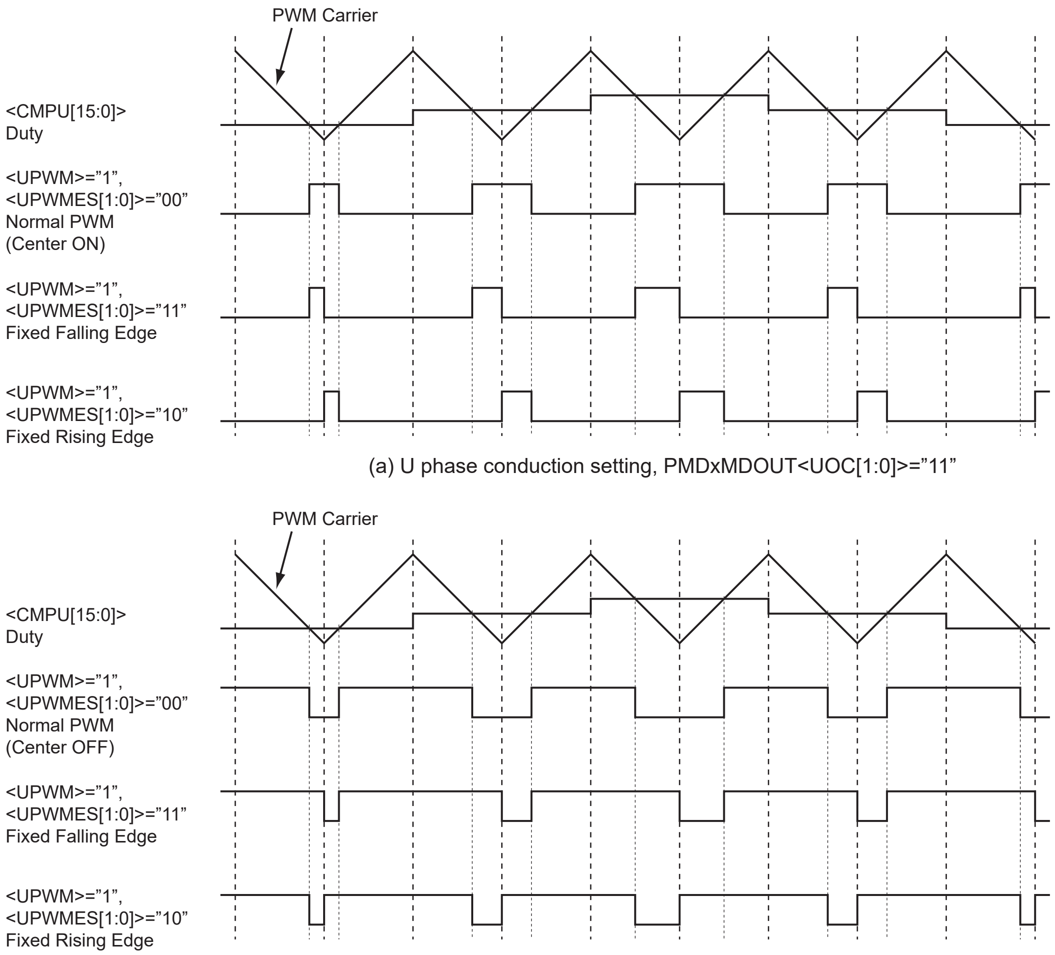

[Sawtooth wave]
<MDCNT>
<MDCNT[15:0]> counts up to PMDxMDPRD and is cleared to "1" in the next cycle.
PMDxMDPRD <MDPRD[15:0]>
PMDxCMPU <CMPU[15:0]>
Time
on
PWMU waveform
off
[Triangular wave]
<MDCNT>
When switching from counting down to counting up, <MDCNT[15:0]>=1 in two cycles.
When switching from counting down to counting up, <MDCNT[15:0]> = peak value (<MDPRD[15:0]>) in two cycle.
PMDxMDPRD <MDPRD[15:0]>
PMDxCMPU <CMPU[15:0]>
Time
on
PWMU waveform
off

Figure 15-5 PWM Waveforms



(a) U phase conduction setting, PMDxMDOUT<UOC[1:0]>="11"

(b) U phase conduction setting, PMDxMDOUT<UOC[1:0]>="00"

Figure 15-6 Waveforms of PWM triangular wave carrier using fixed edge

3. Waveform mode

   Three-phase PWM waveforms can be generated in the following two modes:

   1. 3-phase independent mode:

      Each of the PWM compare registers for the three phases is set independently to generate independent PWM waveforms for each phase. This mode is used to generate drive waveforms such as sinusoidal waves.

   2. 3-phase common mode:

      Only the U-phase PWM compare register is set to generate identical PWM waveforms for all the three phases. This mode is used for rectangular wave drive of brushless DC motors.

4. Interrupt processing

   The pulse width modulation circuit generates PWM interrupt requests in synchronization with PWM waveforms. Interrupt request timing can be selected either at PWM carrier peak or at PWM carrier bottom.

   The PWM interrupt period can be set to half a PWM period, one PWM period, two PWM periods or four PWM periods.

15.2.4.1 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	WPWMES		VPWMES		UPWMES		DSYNCS	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	DTCREN	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-14	WPWMES[1:0]	R/W	W-phase edge setting 00: Edge unfixed (center-aligned PWM) 01: Reserved 10: PWM rising-edge fixed (to the PWM carrier bottom) 11: PWM falling-edge fixed (to the PWM carrier bottom) Note: Valid when triangular carrier PWM is selected (<PWMMD> ="1").
13-12	VPWMES[1:0]	R/W	V-phase edge setting 00: Edge unfixed (center-aligned PWM) 01: Reserved 10: PWM rising-edge fixed (to the PWM carrier bottom) 11: PWM falling-edge fixed (to the PWM carrier bottom) Note: Valid when triangular carrier PWM is selected (<PWMMD> ="1").
11-10	UPWMES[1:0]	R/W	U-phase edge setting 00: Edge unfixed (center-aligned PWM) 01: Reserved 10: PWM rising-edge fixed (to the PWM carrier bottom) 11: PWM falling-edge fixed (to the PWM carrier bottom) Note: Valid when triangular carrier PWM is selected (<PWMMD> ="1").
9-8	DSYNCS[1:0]	R/W	Double buffer update timing for the duty compare register and PWM period register. 00: Depends on interrupt cycle setting (refer to the Table 15-1) Updates at the carrier peak and carrier bottom when 0.5 PWM period is selected (<INTPRD> ="00"). Otherwise, updates at the carrier peak. 01: Updates at PWM carrier bottom 10: Updates at PWM carrier peak 11: Updates at both PWM carrier peak and bottom Note1: Updates at carrier peak when sawtooth wave carrier is selected (<PWMMD>="0") regardless of the setting. Note2: When PMDxMDEN<PWMEN>="0", updates asynchronously regardless of setting.
7	DTCREN	R/W	Set a deadtime correction. 0: Disable 1: Enable
6	PWMCK	R/W	PWM period extension mode 0: Normal period 1: 4 × period Sets the counting cycle of the PWM counter. Normal cycle setting: sawtooth wave 1/fsys (25[ns] at 40[MHz]) / triangular wave 2/fsys (50[ns] at 40 [MHz]) Quadruple cycle setting: sawtooth wave 4/fsys (100[ns] at 40[MHz]) / triangular wave 8/fsys (200[ns] at 40[MHz])
5	SYNTMD	R/W	Port output mode Port outputs are controlled by a combination of <nOC>,<nPWM> and <SYNTMD> (refer to the Table 15-4).

Bit	Bit Symbol	Type	Function
4	DTYMD	R/W	Duty mode 0: 3-phase common mode 1: 3-phase independent mode This bit selects whether to make duty setting independently for each phase or to use the PMDxCMPU register as 3-phase common.
3	PINT	R/W	PWM interrupt request timing 0: Interrupt request occurs at PWM carrier bottom (PMDxMDCNT<MDCNT[15:0]> = 0x0001). 1: Interrupt request occurs at PWM carrier peak (PMDxMDCNT<MDCNT[15:0]> = <MDPRD[15:0]>). Note1: Interrupt request occurs at carrier peak when the PWM carrier is sawtooth wave (<PWMMD>="0"). Note2: Interrupt request occurs both at carrier peak and carrier bottom when the interrupt cycle is 0.5 period (<INTPRD>="00").
2-1	INTPRD[1:0]	R/W	PWM interrupt request cycle 00: Interrupt request at every 0.5 PWM period Note1: PWM interrupt request cycle can be configured only when the PWM carrier is triangular wave (<PWMMD>="1") Note2: The double buffer of the compare registers (PMDxCMPU/V/W) and the cycle register (PMDxMDPRD) are updated by peak and bottom of the PWM carrier. 01: Interrupt request at every PWM period 10: Interrupt request at every two PWM periods 11: Interrupt request at every four PWM periods This field selects the PWM interrupt request period from 0.5 PWM period, one PWM period, two PWM periods and four PWM periods.
0	PWMMD	R/W	PWM carrier waveform 0: PWM mode 0 (edge-aligned PWM and sawtooth wave) 1: PWM mode 1(center-aligned PWM and triangular wave)

Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing

Setting		Update Timing
<DSYNCS[1:0]>	<INTPRD[1:0]>	
00	1x	Updates at PWM carrier peak
	x1	Updates at PWM carrier peak
	00	Updates at PWM carrier peak and PWM carrier bottom
01	xx	Updates at PWM carrier bottom
10	xx	Updates at PWM carrier peak
11	xx	Updates at PWM carrier peak and PWM carrier bottom

x : Don’t care

Table 15-2 Switching control of PMDxCMPU/V/W and VECMPU1/V1/W1

Common settings		Setting for each phase	Register selecting signals
<DSYNCS[1:0]>	<INTPRD[1:0]>	<UPWMES[1]> <VPWMES[1]> <WPWMES[1]>	
01	xx	x	VE register
10	xx	x	VE register
11	xx	0	During up-count: PMD register During down-count: VE register
		1	VE register
00	00	0	During up-count: PMD register During down-count: VE register
		1	VE register

Note: Valid when <MDSEL0>="1", <DCMEN>="1" and <PWMMD>="1".

x: Don’t care

15.2.4.2 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. Note: The PWM carrier is a sawtooth wave (PMDxMDCR<PWMMD>="0"), a zero is always read.

15.2.4.3 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 A value can be read from the up-down counter generating a PWM carrier wave. Counter resolution: 1/fsys (25[ns] at 40[MHz]) Note1: When a quadruple cycle mode is selected (PMDxMDCR<PWMCK>="1"), time resolution of the counter is 4/fsys (100[ns] at 40[MHz]). Note2: Depending on the setting of the PWM carrier (PMDxMDCR<PWMMD>), the PWM counter values when PMD is disabled (PMDxMDEN<PWMEN>="0") are as follows: In case of PMDxMDCR<PWMMD>="0" : 0x0001 In case of PMDxMDCR<PWMMD>="1" : the value of PMDxMDPRD<MDPRD[15:0]>

15.2.4.4 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 By the combination of the PWM period extension mode (PMDxMDCR<PWMCK>) and the PWM carrier waveform <PWMMD>, the PWM cycle can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <MDPRD> × 1/fsys <PWMMD>="1" : <MDPRD> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <MDPRD> × 4/fsys <PWMMD>="1" : <MDPRD> × 8/fsys Note: If <MDPRD[15:0]> is set to a value less than 0x0010, it is automatically assumed to be 0x0010. (The register retains the actual value that is written.)

Note 1: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 2: Since the PMDxMDPRD register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing".

Note 4: Read value is the first buffer value (the latest data set via a bus).

15.2.4.5 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	CMPU							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMPU							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	CMPU[15:0]	R/W	PWM pulse width of U Phase 0x0000 through 0xFFFF Note: When <CMPU> > <MDPRD>, the duty is 100%. <CMPU[15:0]> are compare registers for determining the output pulse width of the U phases. These registers are double-buffered. Pulse width is determined by comparing the subsequent stage buffer and the PWM counter to evaluate which is small or large. By the combination of the PWM period extension mode (PMDxMDCR<PWMCK>) and the PWM carrier waveform (<PWMMD>), the pulse width can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <CMPU> × 1/fsys <PWMMD>="1" : <CMPU> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <CMPU> × 4/fsys <PWMMD>="1" : <CMPU> × 8/fsys

Note 1: To load the subsequent stage buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting PMDxMODESEL<MDSEL0> to "0".

Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 3: Since the PMDxCMPU register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing".

Note 5: Read value is the first buffer value (the latest data set via a bus).

15.2.4.6 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	CMPV							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMPV							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	CMPV[15:0]	R/W	PWM pulse width of V Phase 0x0000 through 0xFFFF Note: When <CMPV> > <MDPRD>, the duty is 100%. <CMPV[15:0]> are compare registers for determining the output pulse width of the V phases. These registers are double-buffered. Pulse width is determined by comparing the subsequent stage buffer and the PWM counter to evaluate which is small or large. By the combination of the PWM period extension mode (PMDxMDCR<PWMCK>) and the PWM carrier waveform (<PWMMD>), the pulse width can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <CMPV> × 1/fsys <PWMMD>="1" : <CMPV> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <CMPV> × 4/fsys <PWMMD>="1" : <CMPV> × 8/fsys

Note 1: To load the subsequent stage buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting PMDxMODESEL<MDSEL0> to "0".

Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 3: Since the PMDxCMPV register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing".

Note 5: Read value is the first buffer value (the latest data set via a bus).

15.2.4.7 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	CMPW							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMPW							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as 0.
15-0	CMPW[15:0]	R/W	PWM pulse width of W Phase 0x0000 through 0xFFFF Note: When <CMPW> > <MDPRD>, the duty is 100%. <CMPW[15:0]> are compare registers for determining the output pulse width of the W phases. These registers are double-buffered. Pulse width is determined by comparing the subsequent stage buffer and the PWM counter to evaluate which is small or large. By the combination of the PWM period extension mode (PMDxMDCR<PWMCK>) and the PWM carrier waveform (<PWMMD>), the pulse width can be calculated as follows: When <PWMCK>="0", <PWMMD>="0" : <CMPW> × 1/fsys <PWMMD>="1" : <CMPW> × 2/fsys When <PWMCK>="1", <PWMMD>="0" : <CMPW> × 4/fsys <PWMMD>="1" : <CMPW> × 8/fsys

Note 1: To load the subsequent stage buffer with the value in the compare register updated via the bus, select the bus mode (default) by setting PMDxMODESEL<MDSEL0> to "0".

Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 3: Since the PMDxCMPW register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing".

Note 5: Read value is the first buffer value (the latest data set via a bus).

15.2.5 Conduction Control Circuit

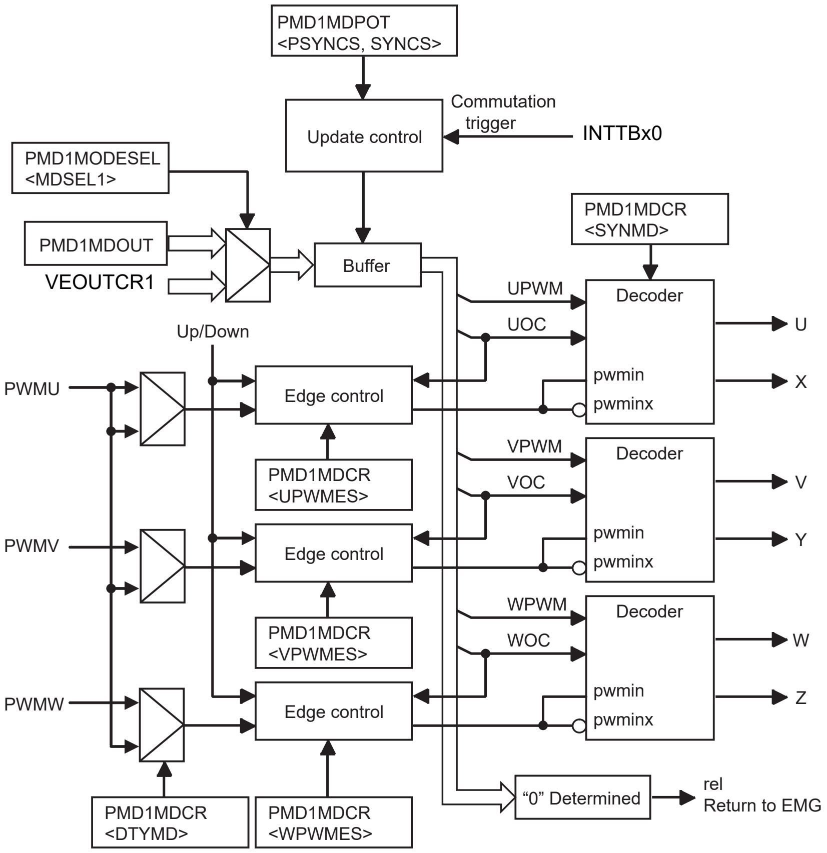


Figure 15-7 Conduction Control Circuit

The conduction control circuit performs the output port control according to the settings made in the output control register PMDxMDOUT(VEOUTCR1) and the output setting register PMDxMDPOT. PMDxMDOUT (VEOUTCR1) register is double-buffered and update timing can be select as synchronous or asynchronous to PWM. Update timing synchronizing with trigger input can also be selected. (For details of update timing, refer to "Table 15-3 Update Timing of the PMDxMDOUT(VEOUTCR1) buffer".)

<WPWM>, <VPWM>, <UPWM> of the PMDxMDOUT(VEOUTCR1) register selects PWM or High/Low output for each of the U, V and W phases. When PWM output is selected, PWM waveforms are output. When High/Low output is selected, output is fixed to either a High or Low level. Each output is set to high or low by <WOC>, <VOC>, <UOC> of the register PMDxMDOUT(VEOUTCR1).

shows port outputs setting according to port output setting in the PMDxMDOUT(VEOUTCR1) register and PMDxMDPOT register, and port output polarity setting in the port output mode of PMDxMDCR register.

Table 15-3 Update Timing of the PMDxMDOUT(VEOUTCR1) buffer

		PSYNCS setting			
		00	01	10	11
SYNCS setting	00	Constant update	PWM carrier bottom	PWM carrier peak	PWM carrier peak and PWM carrier bottom
	01	–	–	–	–
	10	When INTTBx0 is arisen.	The first PWM carrier bottom every time when INTTBx0 is arisen.	The first PWM carrier peak every time when INTTBx0 is arisen.	Either the first PWM carrier peak or the first carrier bottom every time when INTTBx0 is arisen.
	11	–	–		–

Note: IF PMD is disabled (PMDxMDCR<PMWEN>="0"), the retained trigger condition is cleared.

15.2.5.1 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	-	-	-	-	-	-	SYNCS	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	PSYNCS	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-10	−	R	Read as 0.
9-8	SYNCS[1:0]	R/W	Selects PMDxMDOUT(VEOUTCR1) transfer timing (trigger synchronous setting). 00: asynchronous 01: Reserved 10: when INTTBx0 (TMRBx interrupt request) occurs 11: Reserved Selects the subsequent stage buffer update timing of the conduction control register. Note1: By the combination of the settings for <PSYNC> and <SYNCS>, the buffer update timing can be determined (refer to the "Table 15-3 Update Timing of the PMDxMDOUT(VEOUTCR1) buffer"). Note2: When PMD is disabled (PMDxMDEN<PWMEN>="0"), the timing is asynchronous regardless of settings.
7-4	−	R	Read as 0.
3	−	R/W	Always write "0".
2	−	R/W	Always write "0".
1-0	PSYNCS[1:0]	R/W	Selects PMDxMDOUT(VEOUTCR1) transfer timing (PWM synchronous setting). 00: asynchronous to PWM The setting is applied to the port output at the same time that the PMDxMDOUT/VEOUTCR1 registers. 01: Carrier bottom (when <MDCNT[15:0]>="1") 10: Carrier peak (<MDCNT[15:0]>=<MDPRD[15:0]>) 11: Carrier peak and carrier bottom Selects the subsequent stage buffer update timing of the conduction control register. Note1: When the PWM carrier is sawtooth wave, the buffer update timing is the carrier peak, except <PSYNCS>="00" . Note2: By the combination of the settings for <PSYNC> and <SYNCS>, the buffer update timing can be determined (refer to the "Table 15-3 Update Timing of the PMDxMDOUT(VEOUTCR1) buffer"). Note3: When PMD is disabled (PMDxMDEN<PWMEN>="0"), the timing is asynchronous regardless of settings.

Note: This field must be set while PMDxMDEN<PWMEN>="0".

15.2.5.2 PMDxMDOUT(PMD Conduction Control Register)

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

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as 0.
10	WPWM	R/W	W-phase PWM output setting 0: High/Low Output 1: PWM Output Port output is controlled by the combination of <WOC>,<WPWM> and <SYNTMD> (refer to the Table 15-4).
9	VPWM	R/W	V-phase PWM output setting 0: High/Low Output 1: PWM Output Port output is controlled by the combination of <VOC>,<VPWM> and <SYNTMD> (refer to the Table 15-4).
8	UPWM	R/W	U-phase PWM output setting 0: High/Low Output 1: PWM Output Port output is controlled by the combination of <UOC>,<UPWM> and <SYNTMD> (refer to the Table 15-4).
7-6	−	R	Read as 0.
5-4	WOC[1:0]	R/W	W-phase conduction control setting Port output is controlled by the combination of <WOC>,<WPWM> and <SYNTMD> (refer to the Table 15-4).
3-2	VOC[1:0]	R/W	V-phase conduction control setting Port output is controlled by the combination of <VOC>,<VPWM> and <SYNTMD> (refer to the Table 15-4).
1-0	UOC[1:0]	R/W	U-phase conduction control setting Port output is controlled by the combination of <UOC>,<UPWM> and <SYNTMD> (refer to the Table 15-4).

Note 1: To load the subsequent stage buffer with the value in the PMDxMDOUT(VEOUTCR1) register updated via the bus, select the bus mode (default) by setting PMDxMODESEL<MDSEL0> to "0".

Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 3: Since the conduction control register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-3 Update Timing of the PMDxMDOUT (VEOUTCR1) buffer".

Note 5: Read value is the first buffer value (the latest data set via a bus).

Table 15-4 Port Outputs according to the <UOC>,<VOC>,<WOC>,<UPWM>,<VPWM> and <WPWM> settings

PMDxMDCR<SYNTMD>="0"

High-side polarity: Active low
Low-side polarity: Active high

PMDxMDOUT Conduction Control		<WPWM><VPWM><UPWM> PWM output setting			
(Upper phase)	(Lower phase)				
		0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0] ><UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	L	PWM	PWM
0	1	H	H	H	PWM
1	0	L	L	$\overline{\text{PWM}}$	L
1	1	L	H	$\overline{\text{PWM}}$	$\overline{\text{PWM}}$

PMDxMDCR<SYNTMD>=1

High-side polarity: Active low
Low-side polarity: Active high

PMDxMDOUT Conduction Control		<WPWM><VPWM><UPWM> PWM output setting			
(Upper phase)	(Lower phase)				
		0: H/L output		1: PWM output	
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0] ><UOC[0]>	Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	L	PWM	PWM
0	1	H	H	H	$\overline{\text{PWM}}$
1	0	L	L	$\overline{\text{PWM}}$	L
1	1	L	H	$\overline{\text{PWM}}$	$\overline{\text{PWM}}$

15.2.6 Protection Control Circuit

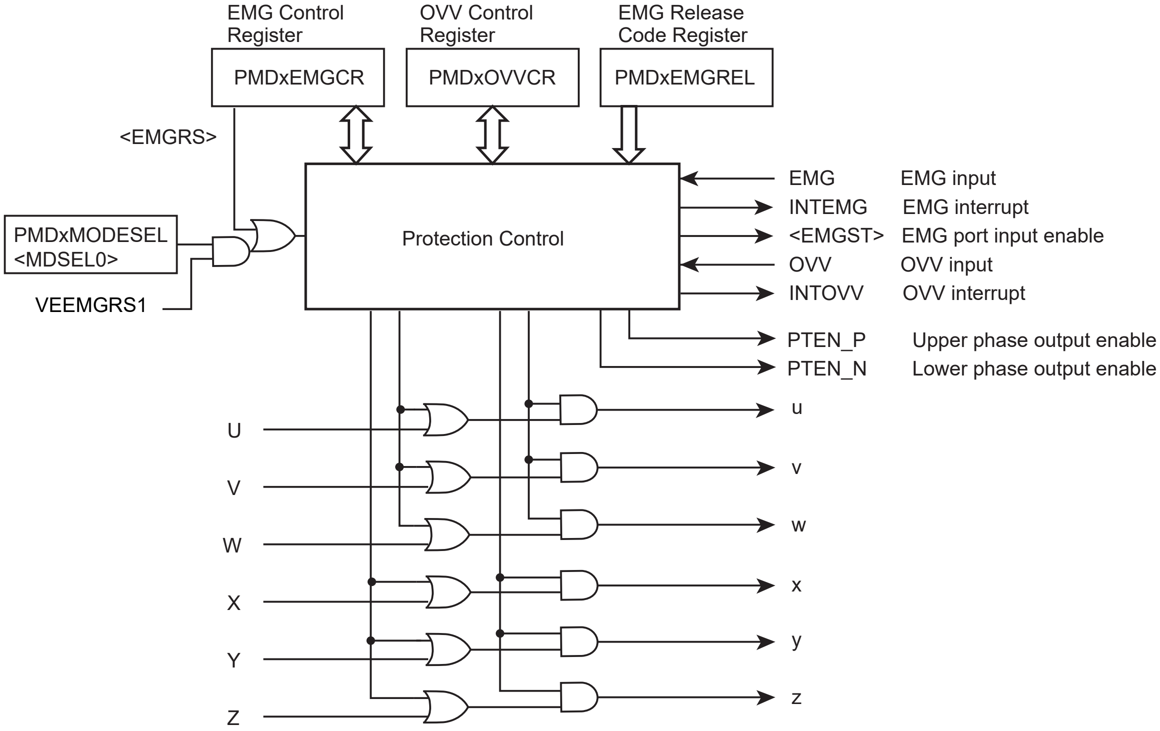


Figure 15-8 Protection Control Circuit

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

15.2.6.1 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 PMDxPORTMD<PORTMD> register. A read value of 1 in EMGSTA<EMGST> indicates that the EMG protection circuit is active.

EMG protection is set through the EMG Control Register (PMDxEMGCR).

In the EMG protection state, it can be released by setting all the port output lines inactive (Set "0" to PMDxMDOUT(VEOUTCR1)<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC>.) (Note1) and then setting either PMDxEMGCR<EMGRS> or VEEMGRS1<EMGRS> to "1". To disable the EMG protection function, write "0x5A" and "0xA5" in this order to the PMDxEMGREL register and then clear PMDxEMGCR<EMGEN> to "0". (These three instructions must be executed consecutively.) While the EMG protection input is low, any attempt to release the EMG protection state is ignored. The EMG protection state can release after that confirming the status flag of PMDxEMGSTA<EMGI> is "1".

The EMG protection circuit can be disabled only after the specified key codes ("0x5A", "0xA5") are written in the PMDxEMGREL register to prevent it from being inadvertently disabled.

Note1: The data of PMDxMDOUT(VEOUTCR1) is necessary to be reflected in the subsequent stage buffer.

Note2: Initial procedure for EMG function

After reset, the EMG function is enabled but EMG pin is configured as a normal port. Therefore, as the EMG protection might be valid, release the EMG protection by the following procedure at the initial sequence.

1: Selects EMG function by PxFR register.

2: Reads PMDxEMGSTA<EMGI> to confirm it as "1".

3: Sets PMDxMDOUT(VEOUTCR1)<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC> to "0" to make all ports in-active.

4: Releases EMG protection by setting PMDxEMGCR(VEEMGRS1)<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.

15.2.6.2 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/OVV disable code The EMG and OVV protection functions can be disabled by setting 0x5A and 0xA5 in this order to register. After writing disable code, set immediately PMDxEMGCR<EMGEN>="0" or PMDxOVVCR<OVVEN>="0". When disabling these functions, <EMGEN> and <OVVEN> must be cleared to "0".

Note: Write a disable code each at disabling EMG and OVV.

15.2.6.3 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 0x0 through 0xF (When <EMGCNT[3:0]>="0", the noise filter is bypassed.) The noise filtering length of anomaly detection input set to these bits. And this value can be calculated by following formula. <EMGCNT[3:0]> × 16/fsys (resolution: 400[ns] at 40[MHz])
7-6	−	R	Read as 0.
5	INHEN	R/W	Tool break enable/disable 0: Disable 1: Enable This bit selects whether or not to stop the PMD when the PMD stop signal is input from the tool. Note: Tool break is enabled in the initial status.
4-3	EMGMD[1:0]	R/W	EMG protection mode select 00: All phases High-Z 01: All upper-phase ON / all lower-phase High-Z 10: All upper phase High-Z / all lower phase ON 11: All phase High-Z Sets the port output both the upper (UO, VO, WO) and the lower (XO, YO, ZO) for the case when EMG occurs. Note: "ON" indicates that PWM output continues.
2	−	R/W	Always write "0".
1	EMGRS	W	EMG protection release 0: − 1: Release protection EMG protection can be released by setting the PMDxMDOUT(VEOUTCR1) register to "0x000" and then setting the <EMGRS> bit to "1". Note: This bit is always read as 0. Note: EMG protection cannot be released if the subsequent stage buffer of PMDxMDOUT(VEOUTCR1) is not updated to "0x000". Note: Before releasing EMG protection, make sure that the PMDxEMGSTA<EMGI> has returned to "1".
0	EMGEN	R/W	EMG protection circuit enable/disable 0: Disable 1: Enable To disable the function, write "0x5A" and then write "0xA5" to the EMG release register(PMDxEMGREL). Then, set "0" to <EMGEN> (These three instructions must be executed consecutively.). Note: This EMG protection circuit is enabled in the initial status.

15.2.6.4 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	Undefined	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	EMGI	R	EMG input EMG protection state The EMG input state can be distinguished by reading this bit
0	EMGST	R	EMG protection state 0: Normal operation 1: Protected The EMG protection state can be distinguished by reading this bit.

15.2.6.5 OVV Protection Control Circuit (OVV Input Block)

The OVV protection control circuit consists of an OVV protection control unit and a port output disable unit. This circuit is activated when the OVV input port is asserted.

When the OVV input signal is asserted (H→L) for a specified period (set to PMDxOVVCR <OVVCNT>), the OVV protection circuit fixes the six port output lines in the conduction control circuit to high or low. At this time, an OVV interrupt (INTOVV) is generated. It is possible to select only the upper phase, lower phase or all phase.

OVV protection is set through the PMDxOVVCR of OVV control register. 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". And after the protection input is canceled, OVV protection is automatically released at a predetermined timing. (The OVV protection state is not released while the OVV protection input is low. The state of this port input can be checked by reading PMDxOVVSTA<OVVI>.)

The OVV protection state is released in synchronization with the PWM period (at the timing when PWM count PMDxMDCNT matches PMDxMDPRD. However if an interrupt on a half cycle of PWM is set, the protection state is released when PWM count is "1" or matches PMDxMDPRD.). To disable the OVV protection function, write "0x5A" and "0xA5" in this order to the PMDxEMGREL of EMG release register 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 PMDxEMGREL register to prevent it from being inadvertently disabled.

15.2.6.6 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 Value: 0x1 through 0xF (If "0" is set, it is handled as "1".) The noise filtering length of OVV input set to these bits. And this value can be calculated by following formula. <OVVCNT[3:0]> × 16/fsys (resolution: 400[ns] at 40[MHz]) Note: <OVVCNT[3:0]> is effective only when port input is selected (<OVVISEL>="0").
7	−	R	Read as 0.
6	ADIN1EN	R/W	ADC monitoring function 1 input enable 0: Disable input 1: Enable input Selects enable/disable signals from ADC monitoring function 1 of the AD converter. If you enable it and select ADC monitoring signal for input (<OVVISEL>="1"), the results of the ADC monitoring function 1 as OVV inputs (if OVV protection is enabled). Note: Refer to the chapter "AD conversion monitoring function" in the "12-bit analog/digital converter" for detailed information about AD conversion monitoring function.
5	ADIN0EN	R/W	ADC monitoring function 0 input enable 0: Disable input 1: Enable input Selects enable/disable signals from ADC monitoring function 0 of the AD converter. If you enable it and select ADC monitoring signal for input (<OVVISEL>="1"), the results of the ADC monitoring function 0 as OVV inputs (if OVV protection is enabled). Note: Refer to the chapter "AD conversion monitoring function" in the "12-bit analog/digital converter" for detailed information about AD conversion monitoring function.
4-3	OVVMD[1:0]	R/W	Selects OVV protection mode 00: No output control 01: All upper phase ON, all lower phase OFF 10: All upper phase OFF, all lower phase ON 11: All phase OFF This field controls the outputs of the upper (UO,VO,WO) and lower(XO,YO,ZO) phases when an OVV condition occurs. Note: "ON" indicates that it's fixed to active output. "OFF" indicates that it's fixed inactive output. Note: If OVV and EMG conditions occur simultaneously, the protection mode settings in the bits of <EMGMD[1:0]> become effective.
2	OVVISEL	R/W	Selects OVV input 0: Port input 1: ADC monitor signal This bit selects whether to use port input or the monitor signal from the ADC as the OVV signal to be input to the protection circuit. Note: When the ADC monitor signal is selected, the setting of OVV input detection time <OVVCNT[3:0] > becomes invalid.(Direct input)
1	OVVRS	R/W	Selects OVV protection state release 0: Disable automatic release of OVV protection state 1: Enable automatic release of OVV protection state Note: If automatic release of OVV protection is enabled, when the state changes to OVV protection after detecting anomaly (OVV input makes a high-to-low transition), the OVV protection state can be automatically released when updating the buffer of PWM cycle register PMDxMDPRD after the OVV input transition to "high". (Refer to the "Table 15-1 PMDxMDPRD, PMDxCMPU/V/W and VECMPU1/V1/W1 Buffer Update Timing")
0	OVVEN	R/W	OVV protection circuit enable/disable 0: Disable 1: Enable Note: To disable the function, write "0x5A" and then write "0xA5" to the EMG release register (PMDxEMGREL). Then, set "0" to <OWEN>. (These three instructions must be executed consecutively.)

15.2.6.7 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	Undefined	0

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1	OVVI	R	OVVI input OVVI state The OVV input state (selected by PMDxOVVCR<OVVISEL>) can be distinguished by reading this bit.
0	OVVST	R	OVV protection state 0: Normal operation 1: In protected The OVV state can be distinguished by reading this bit.

15.2.7 Dead Time Control Circuit

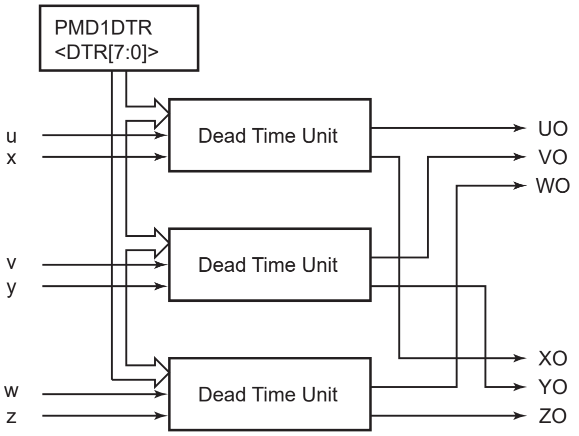


Figure 15-9 Dead Time Control Circuit

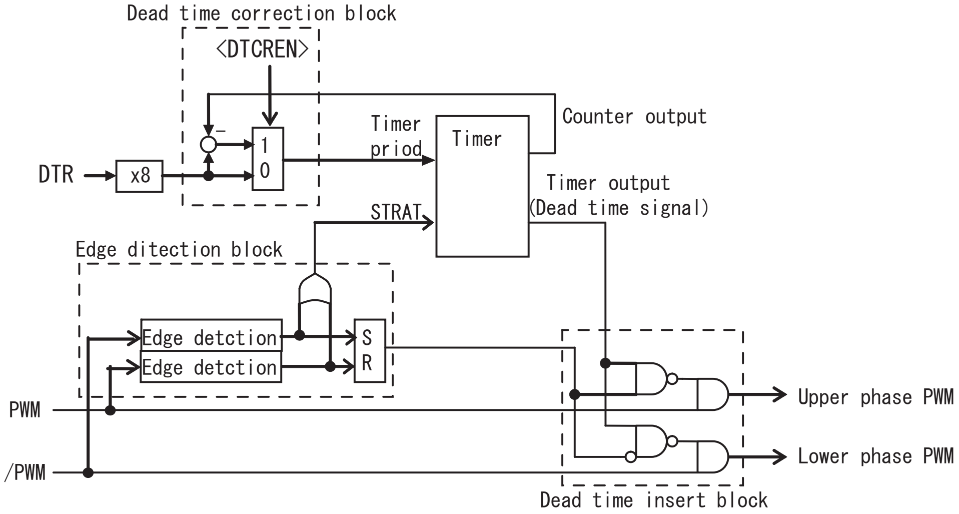


Figure 15-10 The dead time unit configuration

The dead time control circuit consists of a dead time unit and an output polarity switching unit. The dead time unit consists of the edge detection block, timer block, dead time insert block, and dead time correction block. (Refer to "Figure 15-10 The dead time unit configuration")

For each of the U, V and W phases, the dead time units delay the ON-timing of each phase when the upper and lower phases are switched to prevent a short circuit. The dead time is set to the Dead Time Register (PMDxDTR<DTR[7:0]>) as an 8-bit value with a resolution of 8/fsys (200[ns] at 40[MHz]) can be set.

In the dead time correction block, when one of the on-period of the upper PWM or the lower PWM is "0", if PMDxMDCR<DTCREN> is set to "1", the other PWM delay time is corrected to be shortened. If the PWM signal is off during the dead-time period, the delay time of the counter phase is shortened in the rest of dead-time (dead-time register setting time - On time). When the upper PWM becomes OFF during the dead time period, the delay time of the lower PWM should be corrected to be shortened. When the lower PWM becomes OFF during the dead time period, the delay time of the upper PWM should be corrected to be shortened. Figure 15-11 shows dead time correction. A delay time is corrected in the vicinity of duty of 100% of the upper PWM and it also is corrected in the vicinity of duty of 0%.

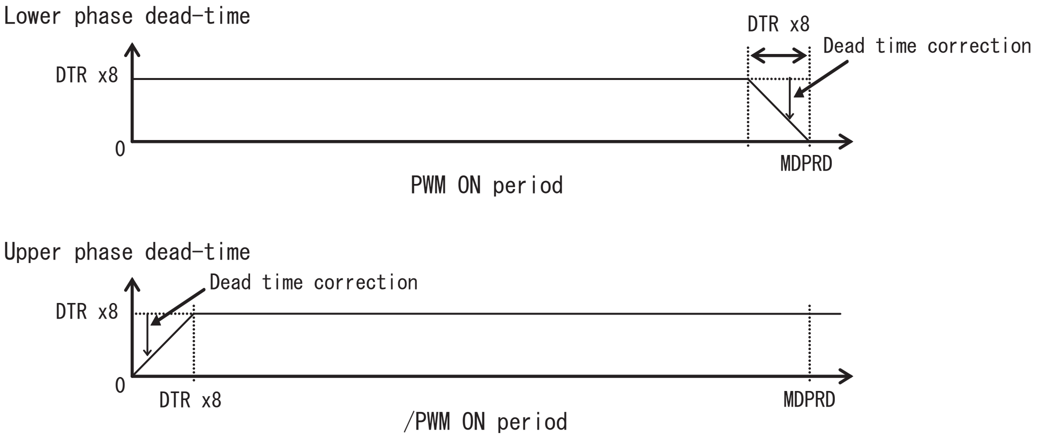


Figure 15-11 Dead Time correction

15.2.7.1 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	Sets Dead time 0x00 through 0xFF The Dead time value can be calculated by following formula. <DTR[7:0]> × 8/fsys (up to 51[μs] at 40[MHz])

15.2.8 Sync Trigger Generation Circuit

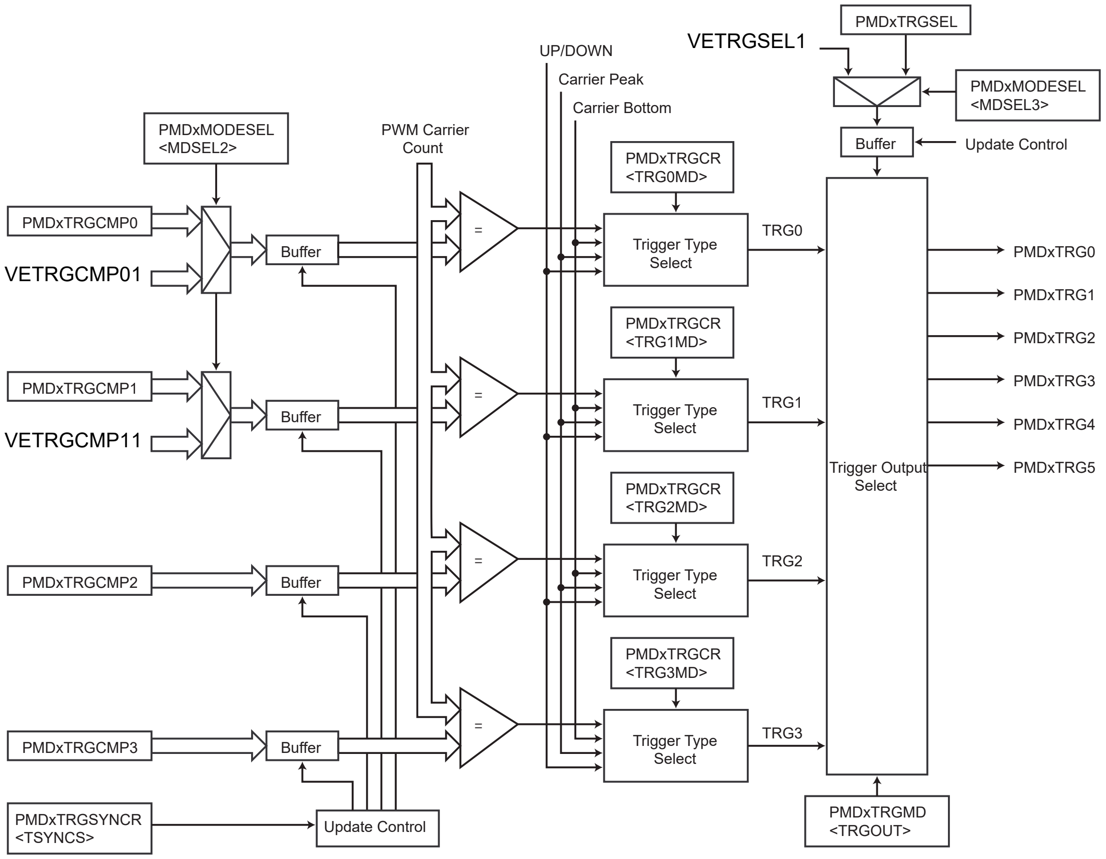


Figure 15-12 Sync Trigger Generation Circuit

The sync trigger generation circuit generates four trigger signals (TRG0 to TRG3) for starting ADC sampling in synchronization with PWM. If VE mode is selected by PMDxMODESEL<MDSEL3>, PMDxTRGCMP0 and PMDxTRGCMP1 become VETRGCMP01, VETRGCMP11 of VE register.

The trigger timing can be selected following 6 types.

1. At up count operation compare-match (Note)
2. At down count operation compare-match (Note)
3. At up-/down count operation compare-match (Note)
4. PWM carrier peak
5. PWM carrier bottom
6. PWM carrier peak and PWM carrier bottom

Note: The compare-match is between PWM counter (PMDxMDCNT<MDCNT[15:0]>) and (PMDxTRGCMPn <TRGCMPn[15:0]>)

During in trigger select output mode: PMDxTRGMD<TRGOUT>="1". The TRG0 signal is output from PMDxTRG0~5 selected by the trigger output select register PMDxTRGSEL (VETRGSEL1). The TRG0 setting is set by PMDxTRGCMP0 (VETRGCMP01) and PMDxTRGCR<TRG0MD>.

When the edge mode (sawtooth wave carrier mode) is selected, the compare-match function is up count. When PMDxTRGMD<EMGTGE>="1", this circuit also outputs trigger signals in EMG protection state.

15.2.8.1 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 register <TRGCMP0[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP0. TRG0 is output. Note: It is prohibited to set <TRGCMP0> to "0" and <TRGCMP0> ≥ <MDPRD[15:0]> value.

Note 1: To load the data in compare registers to the subsequent stage buffer, select the bus mode (default) by setting PMDxMODESEL<MDSEL2> to "0".

Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 3: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-5 Buffer update timing for the trigger compare register".

Note 5: Read value is the first buffer value (the latest data set via a bus).

Table 15-5 Buffer update timing for the trigger compare register

<TSYNCS>setting	<TRGnMD>setting	TRGCMPn register Buffer update timing
00	000	Updates immediately
	001	Update when PWM carrier peak
	010	Update when PWM carrier bottom
	011	Update when PWM carrier peak or PWM carrier bottom (Note1)
	1xx	Updates immediately
01	xxx	Update when PWM carrier bottom
10	xxx	Update when PWM carrier peak
11	xxx	Update when PWM carrier peak or PWM carrier bottom (Note1)

Note: x : Don’t care

Note: Asynchronous update PMDxMDEN<PWMEN>="0" regardless of setting.

Note1: Updates at carrier peak when sawtooth wave carrier is selected (PMDxMDCR<PWMMD>="0") .

15.2.8.2 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 register <TRGCMP1[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP1. TRG1 is output. Note: It is prohibited to set <TRGCMP1> to "0" and <TRGCMP1> ≥ <MDPRD[15:0]> value.

Note 1: To load the data in compare registers to the subsequent stage buffer, select the bus mode (default) by setting PMDxMODESEL<MDSEL2> to "0".

Note 2: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 3: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 4: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-5 Buffer update timing for the trigger compare register".

Note 5: Read value is the first buffer value (the latest data set via a bus).

15.2.8.3 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 register <TRGCMP2[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP2. TRG2 is output. Note: It is prohibited to set <TRGCMP2> to "0" and <TRGCMP2> ≥ <MDPRD[15:0]> value.

Note 1: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 2: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-5 Buffer update timing for the trigger compare register".

Note 4: Read value is the first buffer value (the latest data set via a bus).

15.2.8.4 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 register <TRGCMP3[15:0]> should be set in a range of 1 to [<MDPRD[15:0]> set value − 1]. When the PWM counter value <MDCNT[15:0]> matches the value set in TRGCMP3. TRG3 is output. Note: It is prohibited to set <TRGCMP3> to "0" and <TRGCMP3> ≥ <MDPRD[15:0]> value.

Note 1: Do not write to this register in byte unit. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Note 2: Since the trigger compare register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: For detailed update timing of the subsequent stage buffer, refer to the "Table 15-5 Buffer update timing for the trigger compare register"

Note 4: Read value is the first buffer value (the latest data set via a bus).

15.2.8.5 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	Asynchronous update of the PMDxTRGCMP3<TRGCMP3[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMDxTRGCMP3. 0: Sync update 1: Async update (The value written to PMDxTRGCMP3 is immediately reflected.) Note: For detailed update timing, refer to the "Table 15-5 Buffer update timing for the trigger compare register". Note: When PMDxMDEN<PWMEN>="0", updates asynchronously regardless of setting.
14-12	TRG3MD[2:0]	R/W	PMDxTRGCMP3 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMDxMDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG3MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG3MD[2:0]>. Note: When <TRG3MD[2:0]>="011", PMDxTRGCMP3="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
11	TRG2BE	R/W	Asynchronous update of the PMDxTRGCMP2<TRGCMP2[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMDxTRGCMP2. 0: Sync update 1: Async update (The value written to PMDxTRGCMP2 is immediately reflected.) Note: For detailed update timing, refer to the "Table 15-5 Buffer update timing for the trigger compare register". Note: When PMDxMDEN<PWMEN>="0", updates asynchronously regardless of setting.

Bit	Bit Symbol	Type	Function
10-8	TRG2MD[2:0]	R/W	PMDxTRGCMP2 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMDxMDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG2MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG2MD[2:0]>. Note: When <TRG2MD[2:0]>="011", PMDxTRGCMP2="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
7	TRG1BE	R/W	Asynchronous update of the PMDxTRGCMP1<TRGCMP1[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMDxTRGCMP1. 0: Sync update 1: Async update (The value written to PMDxTRGCMP1 is immediately reflected.) Note: For detailed update timing, refer to the "Table 15-5 Buffer update timing for the trigger compare register". Note: When PMDxMDEN<PWMEN>="0", updates asynchronously regardless of setting.
6-4	TRG1MD[2:0]	R/W	PMDxTRGCMP1 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMDxMDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG1MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG1MD[2:0]>. Note: When <TRG1MD[2:0]>="011", PMDxTRGCMP1="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.
3	TRG0BE	R/W	Asynchronous update of the PMDxTRGCMP0<TRGCMP0[15:0]> buffer This bit enables asynchronous updating to the subsequent stage buffer from PMDxTRGCMP0. 0: Sync update 1: Async update (The value written to PMDxTRGCMP0 is immediately reflected.) Note: For detailed update timing, refer to the "Table 15-5 Buffer update timing for the trigger compare register". Note: When PMDxMDEN<PWMEN>="0", updates asynchronously regardless of setting.

Bit	Bit Symbol	Type	Function
2-0	TRG0MD[2:0]	R/W	PMDxTRGCMP0 mode setting This register selects a match-mode of trigger output. 000: Trigger output disabled 001: Trigger output at down-count match 010: Trigger output at up-count match 011: Trigger output at up-/down-count match 100: Trigger output at PWM carrier peak 101: Trigger output at PWM carrier bottom 110: Trigger output at PWM carrier peak/bottom 111: Trigger output disabled Note: When a "0" is set to PMDxMDCR<PWMMD> (a sawtooth wave), a trigger is output on up-counter match even "001" is set to <TRG0MD[2:0]>. Also, a trigger is output at PWM carrier peak even "101" is set to <TRG0MD[2:0]>. Note: When <TRG0MD[2:0]>="011", PMDxTRGCMP0="0x0001" and PMDxMDCR<PWMMD>="1" (triangular wave), one trigger output is made per period.

15.2.8.6 PMDxTRGSYNCR (Trigger Update Timing Setting Register)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as 0.
1-0	TSYNCS	R/W	Update timing setting for the buffer of the trigger compare register. 00: Updates immediately, PWM carrier peak, bottom and peak or bottom is set for each trigger by setting PMDxTRGCR<TRGxMD>. 01: Update when PWM carrier bottom 10: Update when PWM carrier peak 11: Update when PWM carrier peak or bottom Note: Refer to the "Table 15-5 Buffer update timing for the trigger compare register". Note: Asynchronous update PMDxMDEN<PWMEN>="0" regardless of setting.

15.2.8.7 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 fixed trigger outputs are selected, trigger outputs from PMDxTRG0 to PMDxTRG3 output the trigger signals generated by a match with <TRGCMP0[15:0]> to <TRGCMP3[15:0]> respectively. A PMDxTRG4 and a PMDxTRG5 are not output the trigger signals. When variable trigger output is selected, output signals of the <TRGCMP0[15:0]> are output to one of trigger output from PMDxTRG0 through PMDxTRG5. The trigger output signal is selected by trigger output select register. Note: Refer to the "Table 15-6 Trigger Output Patterns" when variable trigger outputs is selected (<TRGOUT>="1").
0	EMGTGE	R/W	Output enable in EMG protection state 0: Disable trigger output in the protection state 1: Enable trigger output in the protection state This bit enables or disables trigger output in the EMG protection state.

Table 15-6 Trigger Output Patterns

<TRGOUT> Setting	Compare Register	<TRGSEL[2:0]> Setting	Trigger Output
<TRGOUT>="0"	PMDxTRGCMP0	×	PMDxTRG0
	PMDxTRGCMP1		PMDxTRG1
	PMDxTRGCMP2		PMDxTRG2
	PMDxTRGCMP3		PMDxTRG3
<TRGOUT>="1"	PMDxTRGCMP0	0	PMDxTRG0
		1	PMDxTRG1
		2	PMDxTRG2
		3	PMDxTRG3
		4	PMDxTRG4
		5	PMDxTRG5
	PMDxTRGCMP1	×	No trigger output
	PMDxTRGCMP2	×	No trigger output
	PMDxTRGCMP3	×	No trigger output

15.2.8.8 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 PMDxTRG0 001: Output from PMDxTRG1 010: Output from PMDxTRG2 011: Output from PMDxTRG3 100: Output from PMDxTRG4 101: Output from PMDxTRG5 110: No trigger output 111: No trigger output This field is effective when the variable trigger output mode is selected (PMDxTRGMD<TRGOUT>="1"). And an output trigger can be selected by setting the PMDxTRGCMP0 register. (Refer to the Table 15-6.)

Note 1: To load the data of the compare register updated via a bus to the subsequent stage buffer, set bus mode (default) by writing "0" to the PMDxMODESEL<MDSEL3>.

Note 2: Since the trigger output selecting register is double buffered, the cycle can be changed during the operation of the PWM counter.

Note 3: The update timing of the subsequent stage buffer is as the same as the compare register (PMDxCMPU/V/W).

Note 4: When PMD is disabled (PMDxMDCR<PWMEN>="0"), updates asynchronously.

16. Vector Engine (VE+)

16.1 Overview

16.1.1 Features

The Vector Engine provides the following features:

1. Executes basic tasks for vector control (coordinate transformation, phase transformation and SIN/COS computation).

   Uses fixed-point format data.

   →No need for software to manage the decimal point alignment.

2. Enables interface (output control, trigger generation, input processing) with the motor control circuit (PMD: Programmable Motor Driver) and AD converter (ADC).
   - Converts computation results from fixed-point format to data format usable in the PMD.
   - Generates timing data for interactive operation with the PMD and ADC.
   - Converts AD conversion results into fixed-point format.

3. Calculates current, voltage and rotation speed by using normalized values with respect to their maximum values in fixed-point format.

4. Implements PI control in current control.

5. Implements phase interpolation (integration of rotation speed).

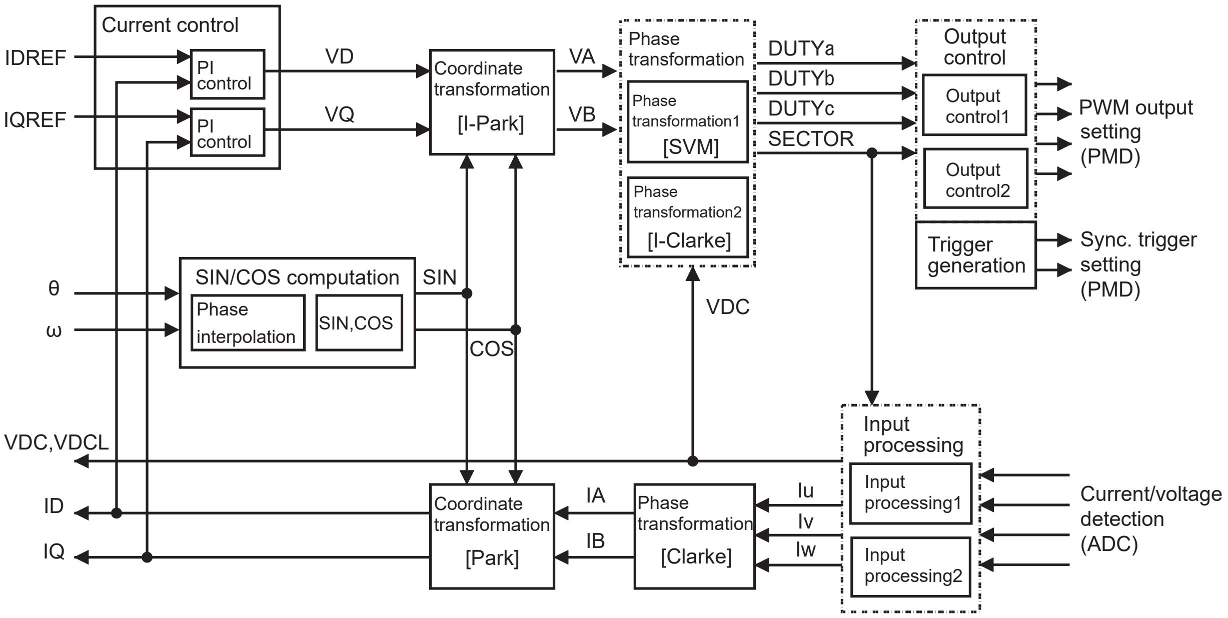


Figure 16-1 Block Diagram of Vector Control

16.1.2 Key Specifications

1. Space vector modulation and inverse Clarke transformation are used for 2-phase-to-3-phase transformation. Space vector modulation supports both 2-phase modulation and 3-phase modulation.
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 set d-axis and q-axis voltage registers.
4. SIN/COS computations are performed with approximations using series expansion. Phase information can be directly specified or computed from rotation speed by using phase interpolation.

Note: It is necessary to set the motor control circuits and the ADC when the Vector Engine to be used.

- For using the Vector Engine, the PMD must be set to the VE mode through the mode select register (PMD1MODESEL).
- It is also necessary to make appropriate settings in the ADC (enabling trigger and selecting AIN and result registers to be used) for each synchronizing trigger from the PMD.

16.2 Configuration

Figure 16-2 shows the configuration of the Vector Engine.

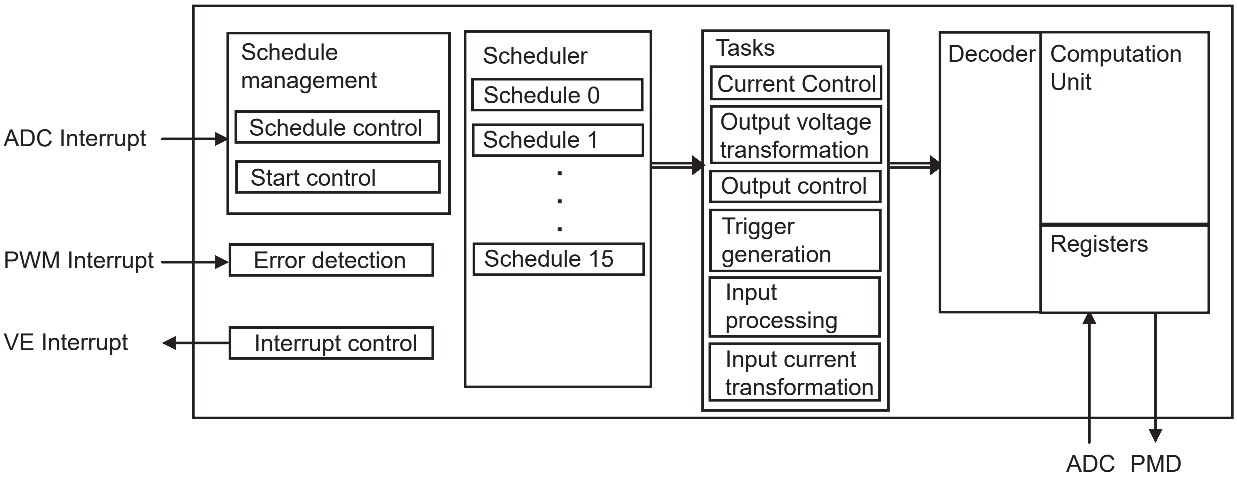


Figure 16-2 Configuration of the Vector Engine

16.2.1 Interaction among Vector Engine, Motor Control Circuit and AD Converter

As shown in Figure 16-3, the Vector Engine allows direct interaction with the PMD and ADC.

When the PMD1MODESEL register is set to the VE mode, the PMD registers PMD1CMPU, PMD1CMPV, PMD1CMPW, PMD1MDOUT, PMD1TRGCMP0, PMD1TRGCMP1 and PMD1TRGSEL are switched to the Vector Engine registers VECMPU1, VECMPV1, VECMPW1, VEOUTCR1, VETRGCMP01,VETRGCMP11 and VETRGSEL1 respectively. In this case, these registers can only be controlled from the Vector Engine, and cannot be written from the PMD. Other PMD registers have no read/ write restrictions.

The registers ADREG0, ADREG1, ADREG2, ADREG3, <UVWISn0>, <UVWISn1>, <UVWISn2>, <UVWISn3> are read into the Vector Engine. (n=0 to 5)

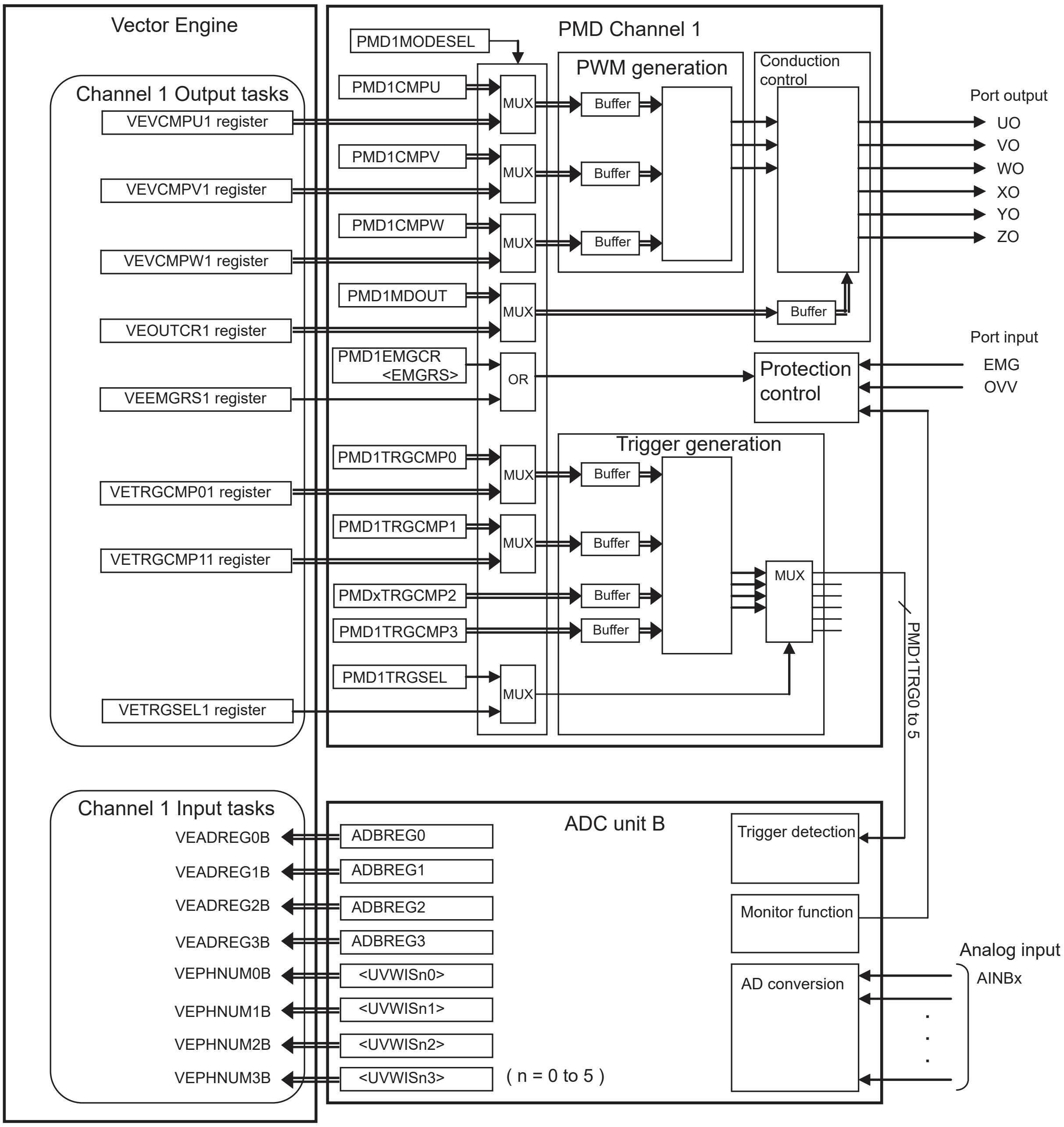


Figure 16-3 Interaction among Vector Engine, PMD and ADC

16.3 List of Registers

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

- VE control registers

  Vector Engine control registers and temporary registers
- Common registers

  Registers common to both channels
- Specific registers

  Computation data and control registers

16.3.1 List of Registers

The table below shows control registers and their addresses.

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

VE Control Registers

Register Name			Address
VE enable/disable	VEEN	R/W	0x0000
CPU start trigger selection	VECPURUNTRG	W	0x0004
Task selection	VETASKAPP	R/W	0x0008
Operation schedule selection	VEACTSCH	R/W	0x000C
Schedule repeat count	VEREPTIME	R/W	0x0010
Start trigger mode	VETRGMODE	R/W	0x0014
Error interrupt enable/disable	VEERRINTEN	R/W	0x0018
VE forced termination	VECOMPEND	W	0x001C
Error detection	VEERRDET	R	0x0020
Schedule executing flag/executing task	VESCHTASKRUN	R	0x0024
Temporary 0	VETMPREG0	R/W	0x002C
Temporary 1	VETMPREG1	R/W	0x0030
Temporary 2	VETMPREG2	R/W	0x0034
Temporary 3	VETMPREG3	R/W	0x0038
Temporary 4	VETMPREG4	R/W	0x003C
Temporary 5	VETMPREG5	R/W	0x0040

Common Registers

Register Name			Address
ADC start wait setting	VETADC	R/W	0x0178

Specific Registers

Register Name			Address
Status flags	VEMCTLF1	R/W	0x00DC
Task control mode	VEMODE1	R/W	0x00E0
Flow control	VEFMODE1	R/W	0x00E4
PWM period rate (PWM period [s] × maximum speed (note1) $\times 2^{16}$) setting	VETPWM1	R/W	0x00E8
Rotation speed (speed [Hz] ÷ maximum speed (note1) $\times 2^{15}$) setting	VEOMEGA1	R/W	0x00EC

Specific Registers

Register Name			Address
Motor phase (motor phase [deg]/360 × 2^{16}) setting	VETHETA1	R/W	0x00F0
d-axis reference value (current [A] ÷ maximum current (note2) ×2^{15}) setting	VEIDREF1	R/W	0x00F4
q-axis reference value (current [A] ÷ maximum current (note2) ×2^{15}) setting	VEIQREF1	R/W	0x00F8
d-axis voltage (voltage [V] ÷ maximum voltage (note3) ×2^{31}) setting	VEVD1	R/W	0x00FC
q-axis voltage (voltage [V] ÷ maximum voltage (note3) ×2^{31}) setting	VEVQ1	R/W	0x0100
Integral coefficient for PI control of d-axis	VECIDKI1	R/W	0x0104
Proportional coefficient for PI control of d-axis	VECIDKP1	R/W	0x0108
Integral coefficient for PI control of q-axis	VECIQKI1	R/W	0x010C
Proportional coefficient for PI control of q-axis	VECIQKP1	R/W	0x0110
Upper 32 bits of integral term (VDI) of d-axis voltage	VEVDIH1	R/W	0x0114
Lower 32 bits of integral term (VDI) of d-axis voltage	VEVDILH1	R/W	0x0118
Upper 32 bits of integral term (VQI) of q-axis voltage	VEVQIH1	R/W	0x011C
Lower 32 bits of integral term (VQI) of q-axis voltage	VEVQILH1	R/W	0x0120
Reference speed to switch PWM shift mode	VEFPWMCHG1	R/W	0x0124
PWM period (to be set identically with PMD’s PWM period)	VEMDPRD1	R/W	0x0128
Minimum pulse width	VEMINPLS1	R/W	0x012C
Synchronizing trigger correction value	VETRGCRC1	R/W	0x0130
DC supply voltage (voltage[V] ÷ maximum voltage (note3) ×2^{15})	VEVDCL1	R/W	0x0134
Cosine value at THETA for output conversion (Q15 data)	VECOS1	R/W	0x0138
Sine value at THETA for output conversion (Q15 data)	VESIN1	R/W	0x013C
Previous cosine value for input processing (Q15 data)	VECOSM1	R/W	0x0140
Previous sine value for input processing (Q15 data)	VESINM1	R/W	0x0144
Sector information	VESECTOR1	R/W	0x0148
Previous sector information for input processing	VESECTORM1	R/W	0x014C
AD conversion result of a-phase zero-current (note4)	VEIAO1	R/W	0x0150
AD conversion result of b-phase zero-current (note4)	VEIBO1	R/W	0x0154
AD conversion result of c-phase zero-current (note4)	VEICO1	R/W	0x0158
AD conversion result of a-phase current (note4)	VEIAADC1	R/W	0x015C
AD conversion result of b-phase current (note4)	VEIBADC1	R/W	0x0160
AD conversion result of c-phase current (note4)	VEICADC1	R/W	0x0164
DC supply voltage (voltage [V] ÷ maximum voltage (note3) ×2^{15})	VEVDC1	R/W	0x0168
d-axis current (current [A] ÷ maximum current (note2) ×2^{31})	VEID1	R/W	0x016C
q-axis current (current [A] ÷ maximum current (note2) ×2^{31})	VEIQ1	R/W	0x0170
PMD control: CMPU setting	VECMPU1	R/W	0x019C
PMD control: CMPV setting	VECMPV1	R/W	0x01A0
PMD control: CMPW setting	VECMPCV1	R/W	0x01A4
PMD control: Output control (MDOUT)	VEOUTCR1	R/W	0x01A8
PMD control: TRGCMP0 setting	VETRGCMP01	R/W	0x01AC
PMD control: TRGCMP1 setting	VETRGCMP11	R/W	0x01B0
PMD control: Trigger selection	VETRGSEL1	R/W	0x01B4
PMD control: EMG return	VEEMGRS1	W	0x01B8

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 converter) × 2^{12}

Note 4: AD conversion results are stored in the upper 12 bits of each 16-bit register.

16.3.2 VE Control Registers

16.3.2.1 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

Note: When the Vector Engine is disabled (VEEN="0"), access to other registers of the Vector Engine is not allowed.

16.3.2.2 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	-
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 Operation starts from the task configured to VETASKAPP<VTASKB>.
0	−	W	Always write "0".

Note 1: When "1" is written to this bit, it is cleared in the next cycle. This bit 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 the schedules and the tasks under executing will be restarted, it must be terminated by VECOMPEND register before a start command executed.

16.3.2.3 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				-	-	-	-
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 1 (Task 0) 0x1: Trigger generation (Task 1) 0x2: Input processing 1 (Task 2) 0x3: Input phase transformation (Task 3) 0x4: Input coordinate axis transformation (Task 4) 0x5: Current control (Task 5) 0x6: SIN/COS computation (Task 6) 0x7: Output coordinate axis transformation (Task 7) 0x8: Output phase transformation 1 [SVM] (Task 8) 0x9: Output control 2 (Task 9) 0xA: Input processing 2 (Task 10) 0xB: Output phase transformation 2 [I-Clarke] (Task 11) 0xC: ATAN (Task 12) 0xD: SQRT (Task 13) 0xE: Reserved 0xF: Reserved Specifies the task to be performed when channel 1 is started by programming.
3-0	−	R/W	Always write "0".

Note: Only those tasks that are included in schedules can be specified.

16.3.2.4 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				-	-	-	-
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 0x2: Schedule 2 0x3: Schedule 3 0x4: Schedule 4 0x5: Schedule 5 0x6: Schedule 6 0x7: Schedule 7 0x8: Schedule 8 0x9: Schedule 9 0xA: Schedule 10 0xB: Schedule 11 0xC: Schedule 12 0xD: Schedule 13 0xE: Schedule 14 0xF: Schedule 15
3-0	−	R/W	Always write "0".

Note: For details, refer to "Table 16-4 Tasks To Be Executed in Each Schedule".

16.3.2.5 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				-	-	-	-
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 repeatedly for a specified number of times
3-0	−	R/W	Always write "0".

Note: When "0" is set, no schedule is executed.

16.3.2.6 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		-	-
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 interrupt 01: Reserved 10: Input process is started by INTADBPDB interrupt (ADCB PMD1 trigger synchronous conversion is complete). 11: Reserved
1-0	−	R/W	Always write "0".

16.3.2.7 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	-
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	−	R/W	Always write "0".

Note: If a PWM interrupt is detected when operation schedule is executing (an activation trigger wait is not included), "1" is set as an error flag.

16.3.2.8 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	-
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	−	W	Always write "0".

Note: When "1" is written to this bit, it is cleared in the next cycle. This bit always read as "0".

16.3.2.9 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	-
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	−	R	Read as "0".

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 reading this register.

16.3.2.10 VESCHTASKRUN (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	-	-	-	-	-
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 1 (Task 0) 0x1: Trigger generation (Task 1) 0x2: Input processing 1 (Task 2) 0x3: Input phase transformation (Task 3) 0x4: Input coordinate axis transformation (Task 4) 0x5: Current control (Task 5) 0x6: SIN/COS computation (Task 6) 0x7: Output coordinate axis transformation (Task 7) 0x8: Output phase transformation 1 [SVM] (Task 8) 0x9: Output control 2 (Task 9) 0xA: Input processing 2 (Task 10) 0xB: Output phase transformation 2 [I-Clarke] (Task 11) 0xC: ATAN (Task 12) 0xD: SQRT (Task 13) 0xE: Reserved 0xF: Reserved
5	VRSCHB	R	Schedule execution status in channel 1 0: Not executing 1: Executing
4-0	−	R	Read as "0".

16.3.2.11 VETMPREG0 (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

16.3.2.12 VETMPREG1 (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

16.3.2.13 VETMPREG2 (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

16.3.2.14 VETMPREG3 (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

16.3.2.15 VETMPREG4 (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

16.3.2.16 VETMPREG5 (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

16.3.3 Common Registers

16.3.3.1 VETADC (ADC Start Wait 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	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	Sets a ADC start wait time. (Refer to Figure 16-4.) 0x0000 to 0xFFFF : (ADC conversion time[s]÷PWM counter clock frequency [s])

Note 1: This register is valid only when 1-shunt current detection mode, PWM shift is enabled and shift 1 are selected.

Note 2: These bits are used by the task 0.

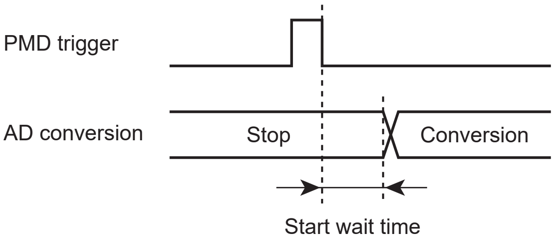


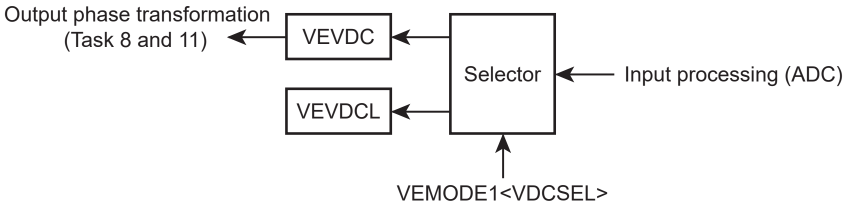
Figure 16-4 ADC start wait

16.3.4 Specific Registers

16.3.4.1 VEMODE1 (Task Control 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	T7SQRTEN	T4ATANEN	-	VDCSEL	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	T7SQRTEN	R/W	Voltage scalar interlocking with task 7: Arithmetic control (SQRT calculation) 0: Disable 1: Enable Refer to the "16.4.2.3 Output Voltage Transformation (Coordinate axis Transformation/Phase Transformation)"
6	T4ATANEN	R/W	Current vector (dq) interlocking with task 4: Phase calculation control (ATAN calculation) 0: Disable 1: Enable Refer to the "16.4.2.7 Input Current Transformation (Phase Transformation/Coordinate axis Transformation)"
5	−	R/W	Always write "0".
4	VDCSEL	R/W	Selects registers to store supply voltage for task 2 and task 10 0: Store VEVDC 1: Store VEVDCL
3-2	OCRMD[1:0]	R/W	Output control for task 0 and task 9. 00: Output OFF 01: Output enable 10: Short circuit brake (output is OFF for the upper phase, ON for the lower phase) 11: Output OFF and EMG return
1	ZIEN	R/W	Zero-current detection for task 2 0: Normal current detection 1: Zero current detection
0	PVIEN	R/W	Phase interpolation control for task 6 0: Disable 1: Enable



Note) When a power supply voltage controlled by VEVDC register is corrected, select VEVDCL register as a storage location, and set a corrected value in VEDVC register.

Figure 16-5 VEVDC/VEVDCL store register

16.3.4.2 VEFMODE1(Flow Control Register)

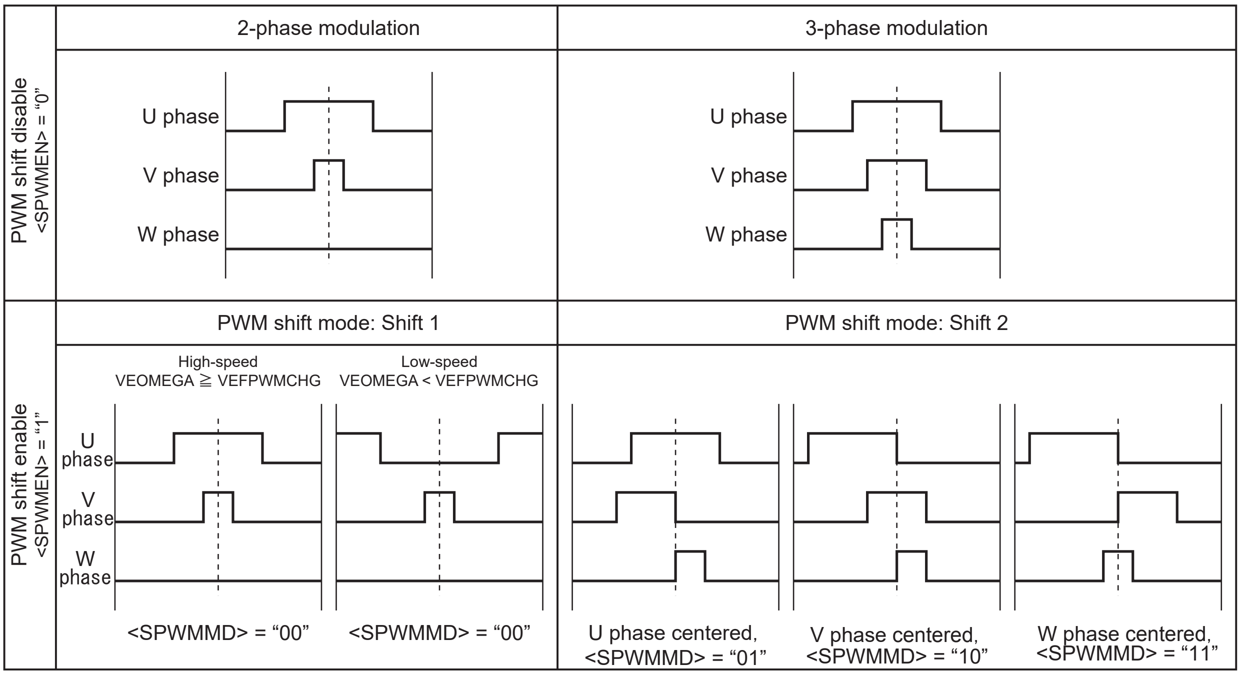
	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SPWMMD		-	PHCVDIS	PIGSEL	-	MREGDIS	CRCEN
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	IDPLMD	-	IDMODE		SPWMEN	C2PEN
After reset	0	1	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-14	SPWMMD	R/W	PWM shift mode selection 00: Shift 1 01: Shift 2 (U-phase centered) 10: Shift 2 (V-phase centered) 11: Shift 2 (W-phase centered) For details, refer to "Figure 16-6 Relation with shift setting and PWM waves". Note: Shift 2 cannot be selected when output control 1 (task 0) is being executed (invalid).
13	−	R/W	Always write "0".
12	PHCVDIS	R/W	Phase transformation disable 0: 2-3 phase transformation enabled (3-phase AC output) 1: 2-3 phase transformation disabled (2-phase AC output) Note: Phase transformation cannot be disabled when space vector modulation (task 8) is running (invalid).
11	PIGSEL	R/W	Switching current control gain 0: Uses each value of PI coefficient register as PI gain. 1: Uses 1/256 of each value of PI coefficient register as PI gain.
10	−	R/W	Always write "0".
9	MREGDIS	R/W	Keep the previous value of SIN/COS/SECTOR 0: Valid 1: Invalid When the invalid setting is chosen, VESINM1=VESIN1, VECOSM1=VECOS1, VESECTORM1=VESECTOR1.
8	CRCEN	R/W	Trigger correction 0: Disable 1: Enable Note: It is effective only when generating a trigger (task 1) and shift disabled in 1-shunt current detection mode or shift 1 is selected.
7-6	−	R/W	Always write "01".
5	IDPLMD	R/W	Current polarity detection 0: Shunt mode (polarity inversion of current input) 1: Sensor mode (no polarity inversion of current input) Note: Valid only when input processing 2 (task 10) is running.
4	−	R/W	Always write "1".

Bit	Bit Symbol	Type	Function
3-2	IDMODE	R/W	Current detection mode 00: 3-shunt (Note 1) 01: 2-sensor (Note 2) 10: 1-shunt (for up count PMD TRG) (Note 3 and Note 4) 11: 1-shunt (for down count PMD TRG) (Note 3 and Note 4) Note 1: Current detection during input processing 2 (task 10) is 3-phase current detection. Note 2: Current detection during input processing 2 (task 10) is 2-phase current detection. Note 4: Set PWM shift 2 if output control 2 (task 9) and input processing 2 (task 10) are executed.
1	SPWMEN	R/W	Enables or disables PWM shift. 0: Disable 1: Enable Note: Output control 1 (task 0) and input processing 1 (task 2) support the shift 1 only. Output control 2 (task 9) and input processing 2 (task 10) support the shift 2 only.
0	C2PEN	R/W	Selects 3-phase or 2-phase modulation. 0: 3-phase modulation 1: 2-phase modulation

Note 3: When 1-shunt mode is used, the acceptable PMDTRG is as follows.

VEFMODE1 <IDMODE[1:0]>	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)



Note1) The shift-1 can be selected 2-phase modulation only.

Note2) The shift-2 can be selected 3-phase modulation only.

Note3) The shift-2 is necessary to set PMD registers.

Figure 16-6 Relation with shift setting and PWM waves

16.3.4.3 VETPWM1(PWM period rate Register)

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

Note: Used for SIN/COS calculation (task 6) when phase interpolation is enabled.

16.3.4.4 VEOMEGA1(Rotation speed Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF: Rotation speed [Hz] ÷ Max_Hz × 2^{15} (Max_Hz: Maximum rotation speed [Hz])

Note 1: It is used for SIN/COS calculation (task 6) when phase interpolation is enabled.

Note 2: It is used when PWM shift 1 of 1-shunt current detection is selected as output control 1 (task 0).

16.3.4.5 VETHETA1(Motor phase Register)

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

Note 1: It is used for SIN/COS calculation (task 6).

Note 2: It is updated to calculate SIN/COS (task 6) when phase interpolation is enabled.

16.3.4.6 VECOS1/VESIN1/VECOSM1/VESINM1(SIN/COS registers)

VECOS1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF

Note 1: It is updated when SIN/COS calculation (task 6).is performed.

Note 2: It is used in output coordinate axis transformation (task 7).

VESIN1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF

Note 1: It is updated when SIN/COS calculation (task 6).is performed.

Note 2: It is used in output coordinate axis transformation (task 7).

VECOSM1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 VECOS1 register Cosine value (previous value): 0x8000 to 0x7FFF

Note 1: It is updated when SIN/COS calculation (task 6).is performed.

Note 2: It is used in input coordinate axis transformation (task 4).

VESINM1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 VESIN1 register Sine value (previous value): 0x8000 to 0x7FFF

Note 1: It is updated when SIN/COS calculation (task 6).is performed.

Note 2: It is used in input coordinate axis transformation (task 4).

16.3.4.7 VEIDREF1/VEIQREF1(d-axis/q-axis Current Reference Registers)

VEIDREF1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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) 0x8000 to 0x7FFF(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}

Note: It is used in current control (task 5).

VEIQREF1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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) 0x8000 to 0x7FFF(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}

Note: It is used in current control (task 5).

16.3.4.8 VEVD1/VEVQ1(d-axis/q-axis Voltage Registers)

VEVD1

	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) 0x8000_0000 to 0x7FFF_FFFF(d-axis voltage÷Max_V×2^{31}) Max_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC)×2^{12}

Note 1: It is updated when current control (task 5) is performed.

Note 2: It is used in output coordinate axis transformation (task 7).

VEVQ1

	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) 0x8000_0000 to 0x7FFF_FFFF(q-axis voltage÷Max_V×2^{31}) Max_V: (Supply voltage (VDC) value [V] which corresponds to 1 LSB of ADC)×2^{12}

Note 1: It is updated when current control (task 5) is performed.

Note 2: It is used in output coordinate axis transformation (task 7).

16.3.4.9 VECIDKI1/VECIDKP1/VEVCIQKI1/VECIQKP1(PI Control Coefficient Registers)

VECIDKI1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF

VECIDKP1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF

VEVCIQKI1

	31	30	29	28	27	26	25	24
bit symbol	-							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF

VECIQKP1

	31	30	29	28	27	26	25	24
bit symbol	-							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: 0x8000 to 0x7FFF

16.3.4.10 VEVDIH1/VEVDILH1/VEVQIH1/VEVQILH1(PI Control Integral Term Registers)

VEVDIH1

	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

VEVDILH1

	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	Lower 16 bits of the integral term (VDI) for PI control of d-axis
15-0	−	R	Read as "0".

Note 1: 64-bit fixed-point data with 63 fractional bits (-1.0 to 1.0)

Note 2: The VDI data is made up of 48 bits.

VEVQIH1

	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

VEVQILH1

	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	Lower 16 bits of the integral term (VQI) for PI control of q-axis
15-0	−	R	Read as "0".

Note 1: 64-bit fixed-point data with 63 fractional bits (-1.0 to 1.0)

Note 2: The VQI data is made up of 48 bits.

16.3.4.11 VEMCTLF1(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> Note: Updated when output control (task 0 and task 9) is performed.
4	PLSLF	R/W	PWM Duty check flag Note: Updated when output control (task 0 and task 9) is performed.
			In the condition that 1-shunt current detection is selected for performing the output control 1(task 0) Minimum disparity of pulse width < VEMINPLS1<MINPLS> case="1" Note: It is disabled in the case low-speed detection(<LAVF> ="1").
			In the condition that output control 2 (task 9) is performing Minimum ON width or minimum OFF width < VEMINPLS1<MINPLS> case ="1"
3	−	R/W	Always write "0".
2	LVTF	R/W	Supply voltage lower flag VEVDC1<VDC> ≥ 0x0100 (1/128) case="0" VEVDC1<VDC> < 0x0100 (1/128) case="1" Note: Updated when output phase (task 5) is performed.
1	LAVFM	R/W	Previous <LAVF> value Note: Updated when output control (task 0 and task 9) is performed.
0	LAVF	R/W	Low-speed flag Note: Updated when output control (task 0 and task9) is performed. 0: High-speed 1: Low-speed VEOMEGA1<OMEGA> ≥ VEFPWMCHG1<FPWMCHG> case="0" VEOMEGA1<OMEGA> < VEFPWMCHG1<FPWMCHG> case="1" Note1: Low-speed is detectable when PWM shift 1 is enabled (FMODE[3]=FMODE[1]=FMODE[0]) and output control 1 (task0) is executed. Note2: It is always set to "0" when output control 2 (task9) is performed.

16.3.4.12 VEFPWMCHG1 (PWM Switching Speed 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	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	Sets the switching speed when PWM shift is enabled. A value to be set to this register is calculated as follows: Switching speed [Hz] ÷ Max_Hz × 2^{15} (Max_Hz: Maximum rotation speed [Hz])

Note: At the task 0, it is used when PWM shift is enabled for 1-shunt current detection and shift 1 is selected.

16.3.4.13 VEMDPRD1(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	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 PMD1MDPRD register.

Note: It is used in performing output control (task0 and task 9) and trigger generation (task 1).

16.3.4.14 VEMINPLS1(Minimum pulse width 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	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 reference value of minimum pulse width disparity (the minimum disparity value of three-phase PWM (VECMPU1, VECMPV1 and VECMPW1)) when PWM shift is enabled at 1-shunt current detection and output control 1 (task 0) is being executed. The calculating formula is as follow. Disparity of pulse width [s] ÷ PWM counter clock period [s] Set the reference value of minimum pulse width (the minimum duty value of three-phase PWM (VECMPU1, VECMPV1 and VECMPW1)) when output control 2 (task 9) is being executed. The calculating formula is as follow. Pulse width [s] ÷ PWM counter clock period [s]

16.3.4.15 VESECTOR1/VESECTORM1(Sector information Register)

VESECTOR1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	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 0xB Indicates the rotation position at the time of output by 12 sectors each having 30 degrees.

Note 1: It is updated when output phase transformation (task 8 and task 11) is performed.

Note 2: t is used in performing output control 1 (task 0).

VESECTORM1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	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	Previous sector information. Value: 0x0 - 0xB Used in input processing.

Note 1: It is updated when output phase transformation (task 8 and task 11) is performed.

Note 2: It is used in performing input processing 1 (task 2).

16.3.4.16 VEIAO1/VEIBO1/VEICO1(Zero-Current Registers)

VEIAO1

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

Note 1: It is updated at input processing 1 (task 2) when zero current detection mode is selected.

Note 2: AD conversion result is stored in the <IAO[15:4]> and the <IAO[3:0]> is always set to "0".

VEIBO1

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

Note 1: It is updated at input processing 1 (task 2) when zero current detection mode is selected.

Note 2: AD conversion result is stored in the <IBO[15:4]> and the <IBO[3:0]> is always set to "0".

VEICO1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: It is updated at input processing 1 (task 2) when zero current detection mode is selected.

Note 2: AD conversion result is stored in the <ICO[15:4]> and the <ICO[3:0]> is always set to "0".

16.3.4.17 VEIAADC1/VEIBADC1/VEICADC1(Current ADC Result Registers)

VEIAADC1

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

Note 1: It is updated at input processing (task 2 and task 10) is performed.

Note 2: AD conversion result is stored in the <IAADC[15:4]> and the <IAADC[3:0]> is always set to "0".

VEIBADC1

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

Note 1: It is updated at input processing (task 2 and task 10) is performed.

Note 2: AD conversion result is stored in the <IBADC[15:4]> and the <IBADC[3:0]> is always set to "0".

VEICADC1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 1: It is updated at input processing (task 2 and task 10) is performed.

Note 2: AD conversion result is stored in the <ICADC[15:4]> and the <ICADC[3:0]> is always set to "0".

16.3.4.18 VEVDC1/VEVDCL1(Supply Voltage Register)

VEVDC1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 0x7FFF The actual voltage value is: VDC value×Max_V value÷2^{15} Max_V: (Supply voltage value [V] which corresponds to 1 LSB of ADC)×2^{12}

Note 1: It is updated at input processing (task 2 and 10) when the register VEMODE1 is selected as "Store VEVDC".

Note 2: It is used when output phase transformation (task 8 and 11) is performed.

VEVDCL1

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-0	VDCL[15:0]	R/W	Supply voltage (16-bit fixed-point data: 0 to 1.0) Value: 0x0000 to 0x7FFF The actual voltage value is: VDCL value × Max_V value ÷ 2^{15} Max_V: (Supply voltage value [V] which corresponds to 1 LSB of ADC)×2^{12}

Note: It is updated at input processing (task 2 and 10) when the register VEMODE1 is selected as "Store VEVDCL".

16.3.4.19 VEID1/VEIQ1(d-axis/q-axis Current Registers)

VEID1

	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: 0x8000_0000 to 0x7FFF_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}

Note 1: It is updated when input coordinate axis transformation (task 4) is performed.

Note 2: It is used in current control (task 5).

VEIQ1

	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: 0x8000_0000 to 0x7FFF_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)$\times 2^{11}$

Note 1: It is updated when input coordinate axis transformation (task 4) is performed.

Note 2: It is used in current control (task 5).

16.3.4.20 VECMPU1 / VECMPV1/ VECMPW1(PWM Duty Register)

VECMPU1

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

Note 1: It is updated when output control (task 0 and 9) is performed.

Note 2: It is used in trigger generation (task 1).

VECMPV1

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

Note 1: It is updated when output control (task 0 and 9) is performed.

Note 2: It is used in trigger generation (task 1).

VECMPW1

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

Note 1: It is updated when output control (task 0 and 9) is performed.

Note 2: It is used in trigger generation (task 1).

16.3.4.21 VEOUTCR1(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 (Note) 01: WO OFF, ZO ON 10: WO ON, ZO OFF 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 (Note) 01: VO OFF, YO ON 10: VO ON, YO OFF 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 (Note) 01: UO OFF, XO ON 10: UO ON, XO OFF 11: UO ON, XO ON (Note) UO and XO are both ON when <UPWM>=1.

Note: It is updated when output control (task 0 and 9) is performed.

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.)

Table 16-1 <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

Table 16-2 <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

Table 16-3 <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

16.3.4.22 VETRGCRC1(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]

Note 1: It is used in trigger generation (task 1).

Note 2: This register is effective when PWM shift is disabled for 1-shunt current detection or shift 1 is selected.

16.3.4.23 VETRGCMP01/VETRGCMP11(Trigger timing setting Register)

VETRGCMP01

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

Note 1: This register is effective when one of the following PMD trigger mode is selected: count-down match, count-up match, or count-up/-down match.

Note 2: It is used in trigger generation (task 1) when PWM shift is disabled for 1-shunt current detection or shift 1 is selected.

VETRGCMP11

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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: This register is effective when one of the following PMD trigger mode is selected: count-down match, count-up match, or count-up/-down match.

Note 2: This register is ineffective when the PMD trigger output mode is set to trigger select output (PMD1TRGMD<TRGOUT>="1").

Note 3: It is used in trigger generation (task 1) when PWM shift is disabled for 1-shunt current detection or shift 1 is selected.

16.3.4.24 VETRGSEL1(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 1: This register is effective when the PMD trigger output mode is set to trigger select output (PMD1TRGMD<TRGOUT>="1").

Note 2: It is updated when trigger generation (task 1) is performed.

16.3.4.25 VEEMGRS1(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	W	PMD setting: EMG return command for returning from the EMG state 0: - 1: EMG return command

Note 1: When "1" is written to this bit, it is cleared in the next cycle. This bit always read as "0".

Note 2: "1" is set when output control (task 0 and 9) is executed in EMG return mode.

16.4 Description of Operations

16.4.1 Schedule Management

Figure 16-7 shows a flowchart for motor control. The Vector Engine makes state transitions according to the schedule and mode settings which are programmed through the relevant registers.

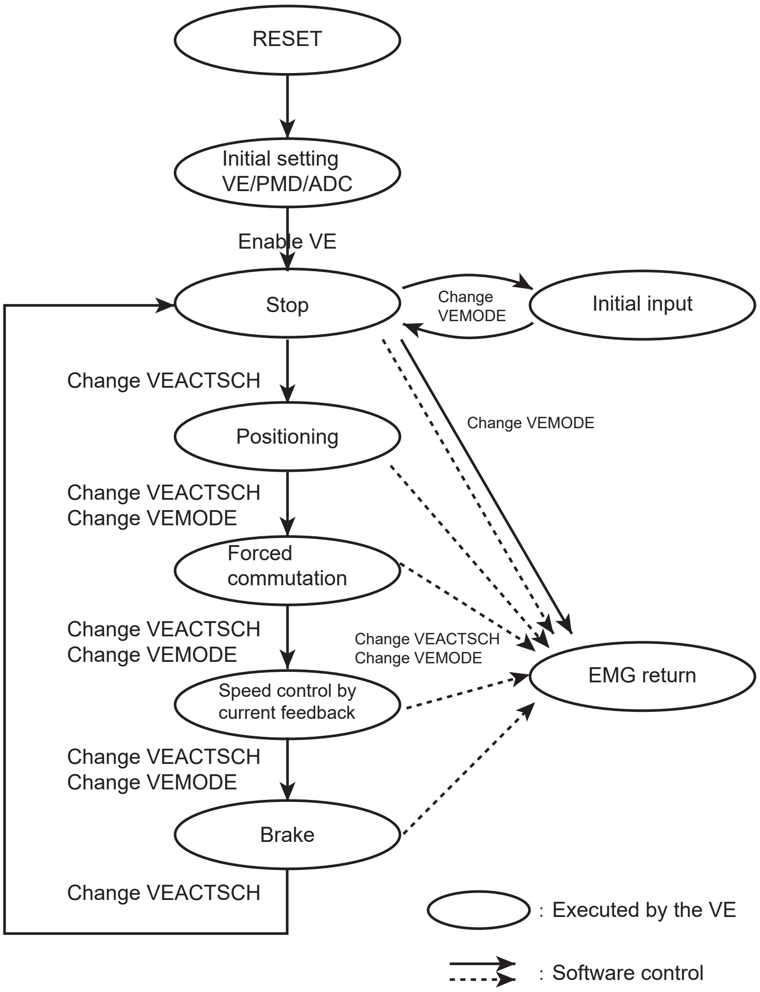


Figure 16-7 Example of Motor Control Flow

RESET	: Microcontroller reset
Initial setting	: Initial setting by a user-created program
Stop	: Stop the motor.
Initial input	: Sample and store zero-current data when the motor is at stop
Positioning	: Determine the initial motor position.
Forced commutation	: Start the motor. For a specified period, the motor is rotated at a specified speed, not controlled by current feedback.
Speed control by current feedback	: Control motor rotation by current feedback.
Brake	: Deceleration control
EMG return	: Return from the EMG state.

16.4.1.1 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 16-4 shows the tasks that are executed in each schedule.

The VEMODE1 register can be selected enable phase interpolation, control output operation, and zero-current detection as appropriate for each step of the motor control flow.

Table 16-4 Tasks To Be Executed in Each Schedule

Schedule Selection VEACTSCH <VACTB[3:0]>	Output Schedule								Input Schedule			
	Current control	SIN/COS computation (Note2)	Output coordinate axis transformation (Note3)	Output phase transformation1	Output phase transformation2	Output control1 (Note4)	Output control2 (Note4)	Trigger generation	Input processing 1 (Note5)	Input processing 2	Input phase transformation	Input coordinate axis transformation (Note6)
	T5	T6	T7	T8	T11	T0	T9	T1	T2	T10	T3	T4
0: Individual task execution	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)
1: Schedule1	o	o	o	o	-	o	-	o	o	-	o	o
2: Schedule2	o	o	o	o	-	-	o	o	-	o	o	o
3: Schedule3	o	o	o	-	o	-	o	o	-	o	o	o
4: Schedule4	-	o	o	o	-	o	-	o	o	-	o	o
5: Schedule5	-	o	o	o	-	-	o	o	-	o	o	o
6: Schedule 6	-	o	o	-	o	-	o	o	-	o	o	o
7: Schedule 7	-	o	o	-	o	o	-	o	o	-	o	o
8: Schedule 8	o	o	o	-	o	o	-	o	o	-	o	o
9: Schedule 9	-	-	-	-	-	o	-	o	o	-	-	-
10: Schedule10	o	o	o	o	-	o	-	-	-	-	-	-
11: Schedule11	o	o	o	o	-	-	o	-	-	-	-	-
12: Schedule12	o	o	o	-	o	-	o	-	-	-	-	-
13: Schedule13	o	o	o	-	o	o	-	-	-	-	-	-
14: Schedule14	-	-	-	-	-	-	-	o	o	-	o	o
15: Schedule15	-	-	-	-	-	-	-	o	-	o	o	o

Note 1: Each task is executed only when it is specified by VETASKAPP.

Note 2: Phase interpolation.

Note 3: When VEMODE1<T7QRTEN> is "1", voltage scalar calculation (SQRT calculation) is performed simultaneously.

Note 4: Output OFF: <EMGRS>

Note 5: Task operation to be switched by zero-current detection.

Note 6: When VEMODE1<T4ATANEN> is "1", current vector (dq) phase calculation (ATAN calculation) is performed simultaneously.

Table 16-5 Typical Setting Example

Register Setting Motor Control Flow	Schedule selection VEACTSCH <VACTB[3:0]>	Task specification VETASKAPP <VTASKB[3:0]>	Phase interpolation VEMODE1 <PVIEN>	Output control VEMODE1 <OCRMD[1:0]>	Zero-current detection VEMODE1 <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
Short brake	4	6	x	10	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.

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.

Note: Schedule 10 to 15 do not support repeating (quits at once even the condition is VEREPTTIME ≥ 2).

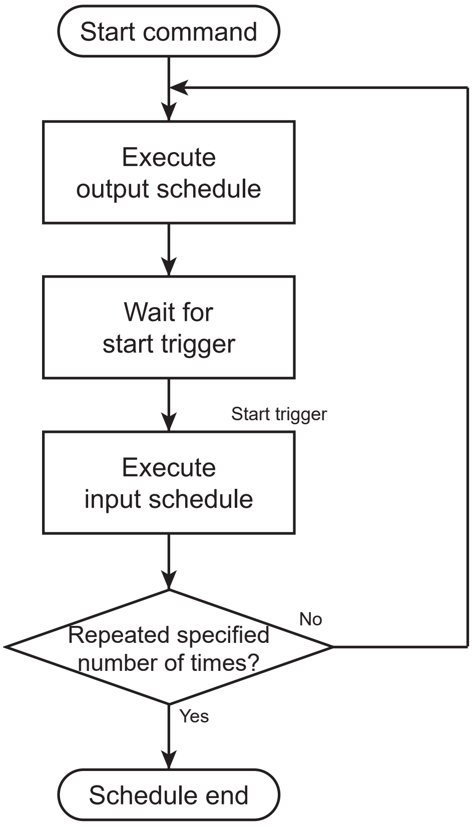


Figure 16-8 Schedule Execution Flow

16.4.1.2 Start Control

Before executing the schedule, set vector engine to enable (<VEEN> ="1"), and then set the operation schedule selection register (VEACTSCH), task specify register (VETASKAPP) and operation schedule repeat specify register (VEREPTIME). The schedule can be executed as follows.

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.

The output schedule and the input it are started by following conditions.

- 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.

16.4.2 Summary of Tasks

Table 16-6 gives a summary of tasks executed in output and input schedules.

When each task is to be executed individually or specified as a startup task, use the task number shown in this table.

Table 16-6 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 transformation	Voltage scalar calculation	7
	Output phase transformation 1	Transforms 2-phase to 3-phase [SVM].	8
	Output phase transformation 2	Transforms 2-phase to 3-phase [I-Clarke].	11
	Output control 1	Converts data to PMD setting format. Switches PWM shift.	0
	Output control 2	Converts data to PMD setting format. Switches PWM shift.	9
	Trigger generation	Generates synchronization trigger timing.	1
Input schedule	Input processing 1	Captures AD conversion results and converts them into fixed-point format.	2
	Input processing 2	Captures AD conversion results and converts them into fixed-point format.	10
	Input phase transformation	Transforms 3-phase to 2-phase.	3
	Input coordinate axis transformation	Current vector (dq) phase calculation	4
ATAN calculation		Calculate arc tangent	12
SQRT calculation		Extract the square root.	13

16.4.2.1 Current Control (Task 5)

The current control task is comprised of a PI control unit for d-axis current and a PI control unit for q-axis current, and calculates d-axis and q-axis voltages.

If current control gain selection (VEFMODE1<PIGSEL>) is set to "1", proportionality and integration constants are multiplied by 1/256.

1. PI control of d-axis current

   <Equations>

Δ ID = VEIDREF1 − VEID1	: Difference between current reference value and current feedback
VEVDI1 = VECIDKI1 × Δ ID + VEVDI1	: Integral term computation
VEVD1 = VECIDKP1 × Δ ID + VEVDI1	: Voltage calculation using proportional term

	Register Name	Function	
Input	VEID1	d-axis current	32-bit fixed-point data (31 fractional bits)
	VEIDREF1	Reference value of d-axis current	16-bit fixed-point data (15 fractional bits)
	VECIDKP1	Proportional coefficient	16-bit data
	VECIDKI1	Integral coefficient	16-bit data
	VEFMODE1	Switching current control gain	<PIGSEL>
Output	VEVD1	d-axis voltage	32-bit fixed-point data (31 fractional bits)
Internal	VEVDI1	Integral term of d-axis voltage	64-bit fixed-point data (63 fractional bits)

Note: The VEVDI is comprised 64 bits. The VEVDIH1 is upper register and the VEVDILH1 is lower it.

2. PI control of q-axis current

   <Equations>

Δ IQ = VEIQREF1 − VEIQ1	: Difference between current reference value and current feedback
VEVQI1 = VECIQKI1 × Δ IQ + VEVQI1	: Integral term computation
VEVQ1 = VECIQKP1 × Δ IQ + VEVQI1	: Voltage calculation using proportional term

	Register Name	Function	
Input	VEIQ1	q-axis current	32-bit fixed-point data (31 fractional bits)
	VEIQREF1	Reference value of q-axis current	16-bit fixed-point data (15 fractional bits)
	VECIQKP1	Proportional coefficient	16-bit data
	VECIQKI1	Integral coefficient	16-bit data
	VEFMODE1	Switching current control gain	<PIGSEL>
Output	VEVQ1	q-axis voltage	32-bit fixed-point data (31 fractional bits)
Internal	VEVQI1	Integral term of q-axis voltage	64-bit fixed-point data (63 fractional bits)

Note: The VEIQ1 is comprised 64 bits. The VEVQIH1 is upper register and the VEVQILH1 is lower it.

16.4.2.2 SIN/COS Computation (Task 6)

The SIN/COS computation task is comprised of a phase interpolation computation and a SIN/COS computation.

Phase interpolation calculates the rotation speed by integrating with the PWM period. It is executed only when phase interpolation is enabled (VEMODE1<PVIEN> ="1").

1. Phase interpolation

   <Equations>

VETHETA1 = VEOMEGA1 × VETPWM1 + VETHETA1	: Integration of rotation speed. Only when phase interpolation is enabled.

	Register Name	Function	
Input	VETHETA1	Phase θ	16-bit fixed-point data (0.0 to1.0, 16 fractional bits)
	VEOMEGA1	Rotation speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VETPWM1	PWM period rate	16-bit data
	VEMODE1	Phase interpolation enable	<PVIEN>
Output	VETHETA1	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)

2. SIN/COS computation

   <Equations>

VESINM1 = VESIN1	: Stores previous value (for input processing).
VECOSM1 = VECOS1	: Stores previous value (for input processing).
VESIN1 = sin (VETHETA1 × π)	: SIN/COS computation
VECOS1 = sin ((VETHETA1 + 1/4) × π)	: SIN/COS computation

	Register name	Function	
Input	VETHETA1	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
OUTPUT	VESIN1	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOS1	Cosine value at θ	
	VESINM1	Previous sine value	
	VECOSM1	Previous cosine value	

16.4.2.3 Output Voltage Transformation (Coordinate axis Transformation/Phase Transformation)

Output voltage transformation is performed in 2 steps; coordinate axis transformation and phase transformation. And phase transformation task consist of 2 types; space vector modulation and inverse Clarke transformation.

1. Output coordinate axis transformation (Task 7)

   In the output coordinate axis task, α-axis and β-axis voltages are calculated using d-axis voltage, q-axis voltage, sinθ and cosθ.

   In the output coordinate axis task, coordinate axis transformation and voltage scalar computation are performed.

   a. Coordinate axis transformation

      <Equations>

      VETMPREG3 = VECOS1 × VEVD1 − VESIN1 × VEVQ1 : Calculates Vα.

      VETMPREG4 = VESIN1 × VEVD1 + VECOS1 × VEVQ1 : Calculates Vβ.

   b. Voltage Scalar computation

      When <T7SQRTEN> is "1", voltage scalar is calculated.

      <Equations>

      VETMPREG5 = SQRT ((VETMPREG3)^2 + (VETMPREG4)^2)

	Register Name	Function	
INPUT	VEVD1	d-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEVQ1	q-axis voltage	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VESIN1	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOS1	Cosine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEMODE1	SQRT execution control	<T7SQRTEN>
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)
	VETMPREG5	Voltage scalar	32-bit fixed-point data (0 to 2.0, 15 fractional bits)

Note: The <VETMPREG5> register is used to store the result when voltage is calculated using scalar operation. This register is used in multiple tasks, so that the calculation result may be overwritten by other tasks. Therefore, when voltage calculation using scalar operation is enabled, perform scalar operation in the individual task execution stage (<VACTB[3:0]>=0x0).

2. Output phase transformation 1 (space vector modulation) (Task 8)

   Output phase transformation 1 determines a sector using α-axis voltage and β-axis. This task is a task in which space vector calculates duties of a-phase voltage, b-phase voltage and c-phase voltage on each task. This task can select either 2-phase or 3-phase modulation as a modulation type.

   a. Determines a sector

<Equations>

```
VESECTORM1 = VESECTOR1                                  : Saves previous sector.
if ($V\alpha \geq 0$ & $V\beta \geq 0$)
    if ($|V\alpha| \geq |V\beta| \div \sqrt{3}$)
        if ($|V\alpha| \div \sqrt{3} \geq |V\beta|$)    SECTOR=0
        else                                            SECTOR=1
    else                                                SECTOR=2
else if ($V\alpha < 0$ & $V\beta \geq 0$)
    if ($|V\alpha| < |V\beta| \div \sqrt{3}$ )          SECTOR=3
    else if ($|V\alpha| \div \sqrt{3} < |V\beta|$)      SECTOR=4
        else                                            SECTOR=5
else if ($V\alpha < 0$ & $V\beta < 0$)
    if ($|V\alpha| \geq |V\beta| \div \sqrt{3}$ )
        if ($|V\alpha| \div \sqrt{3} \geq |V\beta|$)    SECTOR=6
        else                                            SECTOR=7
    else                                                SECTOR=8
else if ($V\alpha \geq 0$ & $V\beta < 0$)
    if ($|V\alpha| < |V\beta| \div \sqrt{3}$ )          SECTOR=9
    else if ($|V\alpha| \div \sqrt{3} < |V\beta|$)      SECTOR=10
        else                                            SECTOR=11
```

	Register Name	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	VESECTOR1	Sector	4-bit data
	VESECTORM1	Previous sector	4-bit data

b. Space vector modulation (indicates <SECTOR[3:0]>="0")

<Equations>

```
if (VESECTOR1<SECTOR[3:0]> = 0)

    $t1 = (\sqrt{3}) \div (VEVDC1) \times ((\sqrt{3}) \div 2 \times V\alpha - 1 \div 2 \times V\beta)$      : Calculates V1 period.
    $t2 = (\sqrt{3}) \div (VEVDC1) \times (V\beta)$                                          : Calculates V2 period.
    $t3 = 1 - t1 - t2$                                                                   : Calculates V0 + V7 period.

if (VEFMODE1<C2PEN> = 0)                                                                 : 3-phase modulation

    $VETMPREG0 = t1 + t2 + t3 \div 2$                                                    : Calculates Va.
    $VETMPREG1 = t2 + t3 \div 2$                                                         : Calculates Vb.
    $VETMPREG2 = t3 \div 2$                                                              : Calculates Vc.

else                                                                                     : 2-phase modulation

    $VETMPREG0 = t1 + t2$                                                                : Calculates Va.
    $VETMPREG1 = t2$                                                                     : Calculates Vb.
    $VETMPREG2 = 0$                                                                      : 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)
	VEVDC1	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VESECTOR1	Sector	4-bit data
	VEFMODE1	Modulation mode setting SECTORM storage setting	<C2PEN> <MREGDIS>
Output	VETMPREG0	a-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)

3. Output Phase Transformation 2 (Inverse Clarke Transformation) (Task 11)

Output phase transformation 2 determines a sector using α-axis and β-axis voltages. This task is a task in which inverse Clarke transformation calculates duties of a-phase, b-phase and c-phase voltages. This task supports only 3-phase modulation as a modulation type.

In addition, if VEFMODE1<PHCVDIS> is set to "1", this task calculates a duty of 2-phase volt-age.

a. Determines a sector

<Equations>

```
VESECTORM1 = VESECTOR1                                              : Stores previous sector.
if (Vα ≥ 0 & Vβ ≥ 0)
    if (|Vα| ≥ |Vβ| ÷ SQRT(3))
        if (|Vα| ÷ SQRT(3) ≥ |Vβ|)              SECTOR=0
        else                                    SECTOR=1
    else                                        SECTOR=2
else if (Vα < 0 & Vβ ≥ 0)
    if (|Vα| < |Vβ| ÷ SQRT(3))                  SECTOR=3
    else if (|Vα| ÷ SQRT(3) < |Vβ|)             SECTOR=4
        else                                    SECTOR=5
else if (Vα < 0 & Vβ < 0)
    if (|Vα| ≥ |Vβ| ÷ SQRT(3))
        if (|Vα| ÷ SQRT(3) ≥ |Vβ|)              SECTOR=6
        else                                    SECTOR=7
    else                                        SECTOR=8
else if (Vα ≥ 0 & Vβ < 0)
    if (|Vα| < |Vβ| ÷ SQRT(3))                  SECTOR=9
    else if (|Vα| ÷ SQRT(3) < |Vβ|)             SECTOR=10
        else                                    SECTOR=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	VESECTOR1	Sector	4-bit data
	VESECTORM1	Previous sector	4-bit data

b. Inverse Clarke Transformation

<Equations>

(VEFMODE1<PHCVDIS> = "0"

$\mathrm{VETMPREG0} = 1 \div \mathrm{VEVDC1} \times V\alpha + 1 \div 2$: Va Duty

$\mathrm{VETMPREG1} = 1 \div \mathrm{VEVDC1} \times (-1 \div 2 \times V\alpha + \mathrm{SQRT}(3) \div 2 \times V\beta) + 1 \div 2$: Vb Duty

$\mathrm{VETMPREG2} = 1 \div \mathrm{VEVDC1} \times (-1 \div 2 \times V\alpha - \mathrm{SQRT}(3) \div 2 \times V\beta) + 1 \div 2$: Vc Duty

(VEFMODE1<PHCVDIS> = "1"

$\mathrm{VETMPREG0} = 1 \div \mathrm{VEVDC1} \times V\alpha + 1 \div 2$: Vα

$\mathrm{VETMPREG1} = 1 \div \mathrm{VEVDC1} \times V\beta + 1 \div 2$: Vβ

	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)
	VEVDC1	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VESECTOR1	Sector	4-bit data
	VEFMODE1	Storing previous value setting Phase transformation disable	<MREGDIS> <PHCVDIS>
Output	VETMPREG0	a-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG1	b-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VETMPREG2	c-phase voltage duty	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)

16.4.2.4 Output Control

The output control unit converts 3-phase voltage duty into PMD setting format. Set the conversion result to VECMPU1, VECMPV1 and VECMPW1, and then set VEOUTCR1 according to the output control setting.

Output control consists of 2 types of task; output control 1 task and output control 2 task. Each task supports different PWM outputs.

1. Output control 1 (Task 0)

   Output control 1 task supports normal PWM outputs and PWM outputs in the shift 1 mode.

   When PWM shift enable is set, if rotational speed (VEOMEGA1) is smaller than the reference of PWM shift switch (VEFPWMCHG1), PWM outputs becomes PWM shifts.

   Note: PWM shift can only be selected in the 1-shunt current detection mode.

	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)
	VEMDPRD1	PWM period	16-bit data (PMD PWM period)
	VESECTOR1	Sector	4-bit data
	VEOMEGA1	Rotation speed	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEFPWMCHG1	Reference speed to switch PWM shift mode	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEMODE1	Output control operation	<OCRMD[1:0]>
	VEMINPLS1	Minimum disparity	16-bit data
	VEFMODE1	Modulation mode PWM shift enable Current detection mode	<C2PEN> <SPWMEN> <IDMODE[1:0]>
Output	VECMPU1	PMD U-phase PMW setting	16- bit data (0 to MDPRD value)
	VECMPV1	PMD V-phase PWM setting	16- bit data (0 to MDPRD value)
	VECMPW1	PMD W-phase PWM setting	16- bit data (0 to MDPRD value)
	VEOUTCR1	PMD output control setting	9-bit setting
	VEEMGRS1	PMD EMG return	1-bit setting
	VEMCTLF1	Low speed flag Small pulse flag	<LAVF> <PLSLF>

2. Output control 2 (Task 9)

Output control 2 task supports normal PWM outputs and PWM outputs in the shift 2 mode.

If PWM shift is enabled (VEFMODE1<SPWMEN> ="1") and PWM shift mode selection (VEFMODE1<SPWMMD>) is set to other than "00", PWM outputs become PWM shift 2 outputs.

Note: PWM shift can only be selected in the 1-shunt current detection mode.

	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)
	VEMDPRD1	PWM period	16-bit data (PMD PWM period)
	VEFPWMCHG1	Reference speed to switch PWM shift mode	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VEMODE1	Output control operation	<OCRMD[1:0]>
	VEMINPLS1	Minimum pulse width	16-bit data
	VEFMODE1	PWM Shift enable Current detection mode PWM shift mode	<SPWMEN> <IDMODE[1:0]> <SPWMMD>
Output	VECMPU1	PMD U-phase PMW setting	16- bit data (0 to MDPRD value)
	VECMPV1	PMD V-phase PWM setting	16- bit data (0 to MDPRD value)
	VECMPW1	PMD W-phase PWM setting	16- bit data (0 to MDPRD value)
	VEOUTCR1	PMD output control setting	9-bit setting
	VEEMGRS1	PMD EMG return	1-bit setting
	VEMCTLF1	Low speed flag Small pulse flag	<LAVF> <PLSLF>

16.4.2.5 Trigger Generation

The trigger generation unit calculates the trigger timing from the PWM setting values (VECMPU1, VECMPV1 and VECMPW1) as appropriate to the current detection method, and sets the VETRGCMP01 and VETRGCMP11 registers.

Note: VETRGCMP01 and VETRGCMP11 are updated only by 1-shunt current detection.

Note: VETRGCMP01 and VETRGCMP11 are not updated when PWM shift 2 mode is selected.

	Register Name	Function	
Input	VECMPU1	PMD U-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPV1	PMD V-phase PWM setting	16-bit data (0 to MDPRD value)
	VECMPW1	PMD W-phase PWM setting	16-bit data (0 to MDPRD value)
	VEMDPRD1	PWM period setting	16-bit data (PMD PWM period)
	VETADC	AD conversion time	16-bit data (0 to MDPRD value)
	VETRGCRC1	Trigger correction value	16-bit data (0 to MDPRD value)
	VESECTOR1	Sector	4-bit data
	VEMODE1	Zero-current detection Output control operation	<ZIEN> <OCRMD[1:0]>
	VEFMODE1	Modulation mode PWM shift enable Current detection mode Trigger correction enable Shift PWM mode	<C2PEN> <SPWMEN> <IDMODE[1:0]> <CRCEN> <SPWMMD>
	VEMCTLF1	Low speed flag	<LAVF>
Output	VETRGCMP01	PMD trigger 0 timing	16-bit data (0 to MDPRD value)
	VETRGCMP11	PMD trigger 1 timing	16-bit data (0 to MDPRD value)
	VETRGSEL1	PMD trigger selection	3-bit data

16.4.2.6 Input Processing

In the input process, Vector Engine reads a conversion result and phase information from AD convertor. Depending on the current detection type and shift PWM mode setting, it converts 3-phase current and voltage of conversion result into the fixed point data and stores them. In zero-current detection mode, current detection results are stored in the zero-current register.

Input process consists of 2 types of task; input process 1 task and input process 2 task. Each task supports different current detection types.

1. Input process 1 (task 2)

   Input process 1 task supports 3-shunt current detection (only 2-phase current are detected (Note 1)) and 1-shunt current detection.

   However, in PWM shift 2 mode, it does not support 1-shunt current detection when PWM outputting (Note 2).

   Note1: Only two phase current among 3-phase current are detected. The rest is calculated using the result of the 2-phase current.

   Note2: PWM shift can only be used in the 1-shunt current detection mode.

	Register Name	Function	
Input	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	
	VEPHNUM0B	ADREG0B detected phase information	2-bit data
	VEPHNUM1B	ADREG1B detected phase information	
	VEPHNUM2B	ADREG2B detected phase information	
	VEPHNUM3B	ADREG3B detected phase information	
	VESECTORM1	Sector information	4-bit data
	VEMODE1	Zero-current detection VDC store register	<ZIEN> <VDCSEL>
	VEFMODE1	Current detection mode	<IDMODE[1:0]>
	VEMCTLF1	Low speed flag	<LAVFM>
Output	VEVDC1	DC power supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VEVDCL1	DC power 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	VEIAO1	a-phase zero-current conversion result	16-bit data (The upper 12 bits are used.)
	VEIBO1	b-phase zero-current conversion result	
	VEICO1	c-phase zero-current conversion result	
	VEIAADC1	a-phase current conversion result	16-bit data (The upper 12 bits are used.)
	VEIBADC1	b-phase current conversion result	
	VEICADC1	c-phase current conversion result	

2. Input process 2 (task 10)

   Input process 2 task supports 3-shunt current detection (3-phase detection, 2-phase detection) and 2 sensor current detection. It also supports 1-shunt current detection when PWM outputting in the PWM shift 2 mode. (Note)

   Input process 2 task does not support zero-current detection mode.

   Note: PWM shift can only be used in the 1-shunt current detection mode.

	Register Name	Function	
Input	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	
	VEPHNUM0B	ADREG0B detected phase information	2-bit data
	VEPHNUM1B	ADREG1B detected phase information	
	VEPHNUM2B	ADREG2B detected phase information	
	VEPHNUM3B	ADREG3B detected phase information	
	VEMODE1	VDC store register	<VDCSEL>
	VEFMODE1	Current detection mode Current detection polarity	<IDMODE[1:0]> <IDPLMD>
Output	VEVDC1	DC power supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VEVDCL1	DC power 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	VEIAADC1	a-phase current conversion result	16-bit data (The upper 12 bits are used.)
	VEIBADC1	b-phase current conversion result	
	VEICADC1	c-phase current conversion result	

16.4.2.7 Input Current Transformation (Phase Transformation/Coordinate axis Transformation)

Input current transformation consists of two tasks; phase transformation and coordinate axis transformation.

1. Input phase transformation (task 3)

   Input phase transformation task calculates Iα and Iβ from Ia, Ib and Ic.

   <Equations>

   VETMPREG3 = VETMPREG0 : Calculates Iα.

   VETMPREG4 = 1 ÷ SQRT(3) × VETMPREG1 − 1 ÷ SQRT(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)
	VEFMODE1	Phase transformation disable	<PHCVDIS>
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. Input coordinate axis transformation (task 4)

   Input coordinate axis transformation task calculates Id and Iq from Iα, Iβ, VESINM1 and VECOSM1

   a. Coordinate axis conversion

      <Equations>

      VEID1 = VECOSM1 × VETMPREG3 + VESINM1 × VETMPREG4 : Calculates Id.

      VEIQ1 = − VESINM1 × VETMPREG3 + VECOSM1 × VETMPREG4 : Calculates Iq.

   b. Current vector phase calculation

      When <T4ATANEN> is "1", phase calculation is performed.

      <Equations>

      VETMPRG5 = ATAN(VEID1 / VEIQ1)

	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	
	VESINM1	Sine value at θ	16-bit fixed-point data (-1.0 to 1.0, 15 fractional bits)
	VECOSM1	Cosine value at θ	
	VEMODE1	ATAN execution enable	<T4ATANEN>
	VEMCTLF1	Small pulse flag	<PLSLFM>
Output	VEID1	d-axis current	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
	VEIQ1	q-axis current	
	VETMPREG5	ATAN result	32-bit data (0xFFFFE000 to 0x00002000)

16.4.2.8 Miscellaneous Tasks

1. ATAN (Arc tangent) (Task 12)

   ATAN task outputs a phase of −45° to 45° calculated in the arc tangent using an input of -1.0 to 1.0.

   <Equations>

```
VETMPREG5 = ATAN(VETMPREG5)                              : ATAN calculation
```

	Register Name	Function	
Input	VETMPREG5	Input value	32-bit fixed-point data (-1.0 to 1.0, 31 fractional bits)
Output	VETMPREG5	Phase value	32-bit data (0xFFFFE000 ~ 0x00002000 (−45° to 45°))

2. SQRT (Square root function) (Task 13)

   SQRT task outputs a value of 0.0 to 2.0 calculated in the square root function using an input of 0.0 to 4.0.

   <Equations>

```
VETMPREG5 = SQRT(VETMPREG5)                              : Square root calculation
```

	Register Name	Function	
Input	VETMPREG5	Input value	32-bit fixed-point data (0.0 to 4.0, 15 fractional bits)
Output	VETMPREG5	Square value	32-bit fixed-point data (0.0 to 2.0, 15 fractional bits)

16.5 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 16-7 Combination of VE and PMD

Vector Engine	PMD
Channel 1	Channel 1

Table 16-8 Combination of VE and ADC

Vector Engine		ADC Unit B			
Channel	Current detection VEMODE1 <IDMODE[1:0]>	ADREG0	ADREG1	ADREG2	ADREG3
1	0x	Current detection 1	Current detection 2	Current detection 3	VDC detection
	1x	Current detection 1	Current detection 2	Current detection 3 (Note)	VDC detection

Note: Current detection 3 is used only PWM shift 2.

17. Op-Amps (AMP)

The TMPM37AFSQG has a op-amp . the op-amp amplifier a voltage received via an input port and feeds its output voltage into a 12-bit successive-approximation analog-to-digital (AD) converter(s). This op-amp is used to amplify voltage differentials across shunt resistors for motor current measurement.

17.1 Configuration

Figure 17-1 shows the block diagram of the op-amps/analog converters.

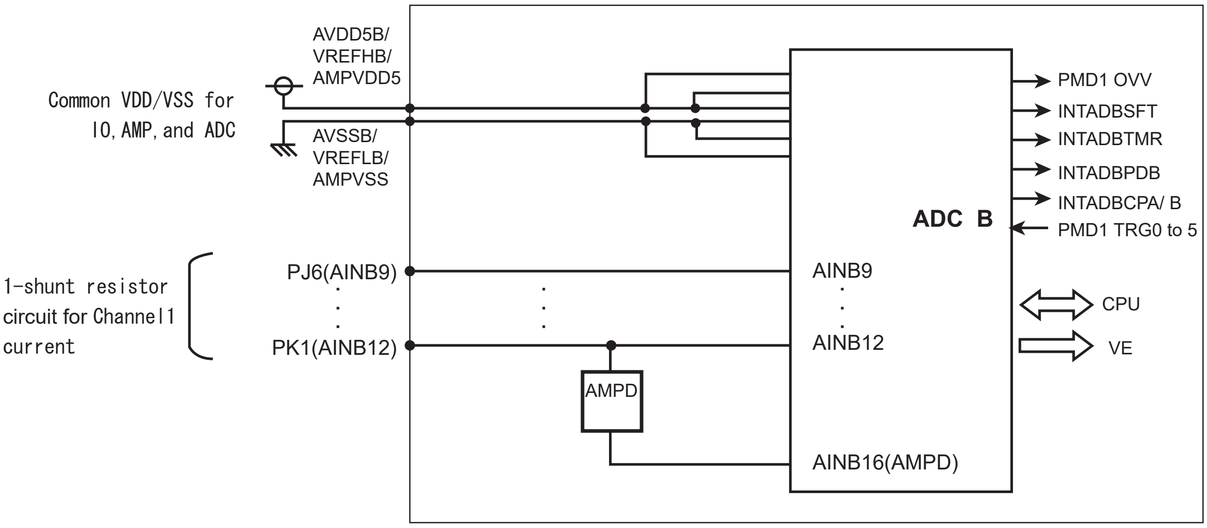


Figure 17-1 Op-Amps/Analog Converters Block Diagram

17.2 Register List

The op-amp is programmable through the AMPCTLD, which allow software to enable and disable the op-amp and select a voltage gain from eight levels.

If an external op-amp is used instead of an on-chip op-amp, the on-chip op-amp should be disabled (<AMP-EN> = 0) .

The following describes the control register.

17.2.1 Op-amps

Base Address = 0x4003_0400

Register		Address(Base+)
Amp D Control Register	AMPCTLD	0x0018

17.2.1.1 AMPCTLD (Amp D 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	-	-	-	-	AMPGLIN			AMPEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as 0.
3-1	AMPGLIN[2:0]	R/W	Gain Select 000: 1.5x 100: 4.0x 001: 2.5x 101: 6.0x 010: 3.0x 110: 8.0x 011: 3.5x 111: 10.0x
0	AMPEN	R/W	AMP Enable 0: Disable 1: Enable

Note: When <AMPEN> is set to "1", it takes approx. 10μs to stabilize the circuit.

17.3 Operation

17.3.1 Basic Operation

Op-amp D (AMP D) only supports 1-shunt current sensing. The amplified voltage from AMP D is fed into one AD converter (AINB16).

See the block diagram of the op-amp shown in Figure 17-2.

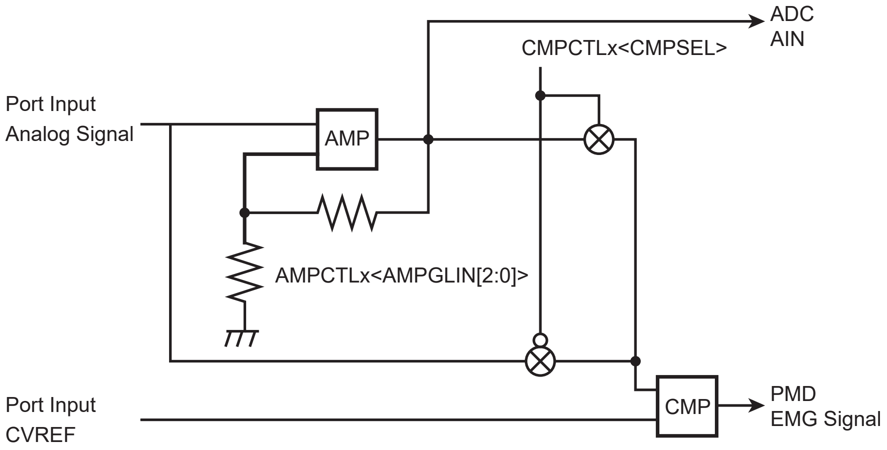


Figure 17-2 Op-Amp

18. Pre-driver(MCD)

3-Phase Full-Wave Pre-Driver for FAN

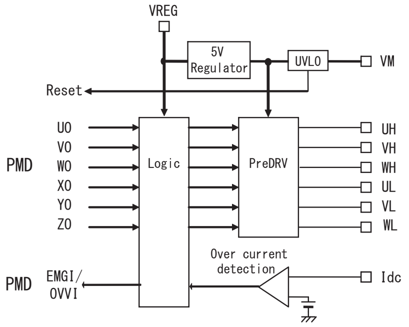


Figure 18-1 Block Diagram of MCD

18.1 Pin Description

Pin name	Input/Outpit	
VM	-	Power supply pin for the MCD. VM = 6V to 32V
IDC	Input	Over current detection input pin
VREG	-	5V pin for the MCD VREG must be connected to GND through 4.7μF capacitor for built-in 5V regulator.
UL	Output	U-phase low side output pin.
UH	Output	U-phase high side output pin.
VL	Output	V-phase low side output pin.
VH	Output	V-phase high side output pin.
WL	Output	W-phase low side output pin.
WH	Output	W-phase high side output pin.

18.2 Functional Description

18.2.1 Pre Driver

Output pins for high side drive are low-active. High side drive is constant-current circuit.

Output pins for low side drive are high-active. Low side drive is push-pull driver circuit.

High-side/low-side arm shorting mode is disabled by the input logic. In this case, all of outputs are turned off.

18.2.2 Under voltage lockout protection (UVLO)

The power supply voltage of VM is always monitored. When it falls to 3.9 V (typ.) or less, it is recognized as low voltage and the circuit is turned off. The normal operation resumes when the voltage recovers to 4.2V (typ.) or more. In normal operation. During low voltage detection, MCU is reset.

18.2.3 Built-in 5V regulator

In the case of 12V motor drive, the built-in regulator can power to the MCU. In this case, the voltage of VM must not exceed 15V.

18.2.4 Over current detection

This function operates when the voltage of IDC pin reaches the detection voltage (0.15 V (typ.)).

Protection operation is determined by the protection circuit (EMG/OVV) of 3-phase PWM timer (PMD).Therefore

19. Power-on-Reset Circuit (POR)

The power-on-reset circuit (POR) generates a power-on reset signal when power-on.

Power supply voltage is indicated as DVDD5.

19.1 Structure

Power-on-reset circuit consists of the reference voltage generation circuit, comparators, the VLTD reset circuit and the power-on counter.

This circuit compares a voltage divided by the ladder resistor with a reference voltage generated in the reference voltage generation circuit in the comparator.

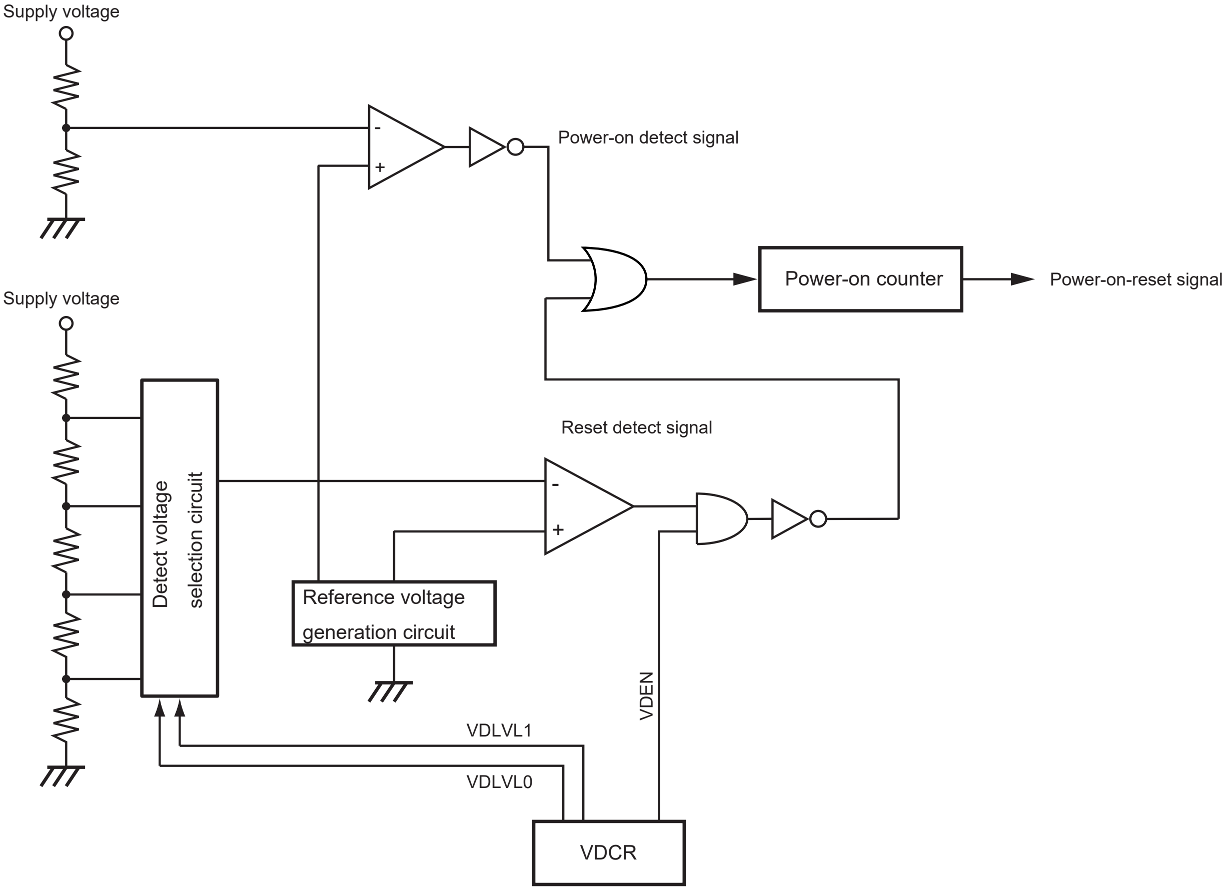


Figure 19-1 Power-on-reset circuit

For details of VDCR in VLTD reset circuit, refer to Section "Voltage detection circuit (VLTD)".

19.2 Function

At power-on, a power-on detection signal generates while power supply voltage is lower than the releasing voltage. Power-on detection signal is released at the timing when DVDD5 is over 3.0 ±0.2 V.

If the power-on detection signal is released and the reset detection signal is also released, the power-on counter starts to operate. After waiting time (approximately 3.2 ms) has elapsed, the power-on reset signal is released.

During the power-on reset signal generation, the CPU and the peripheral functions are reset.

When a reset input is not used, supply voltage must be raised to the recommended operational voltage range until the power-on reset releasing. If power supply voltage does not reach to the recommended operational voltage range during this period, TMPM37AFSQG cannot operate properly.

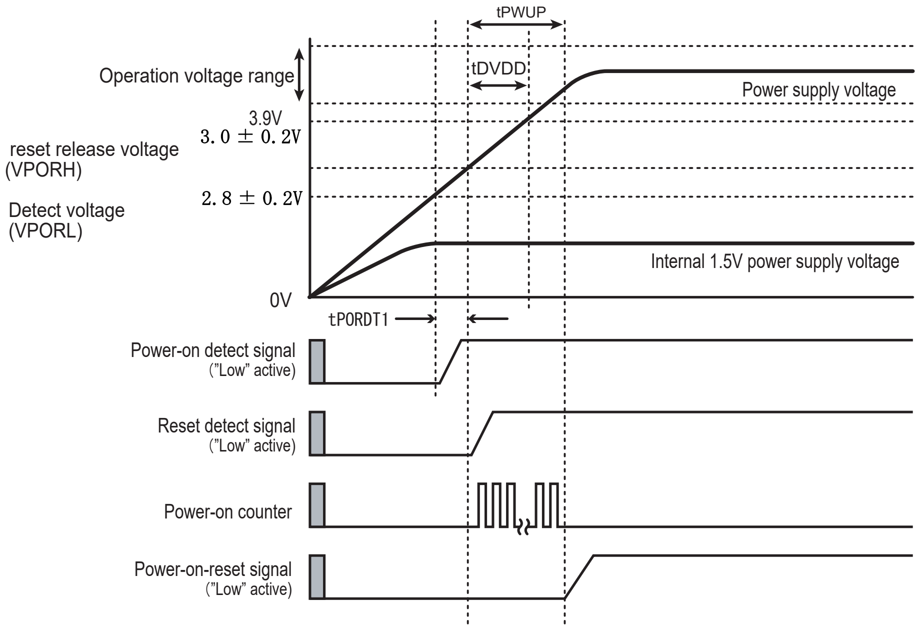


Figure 19-2 Power-on-reset operation timing

Symbol	Parameter	Min	Typ	Max	Unit
tPWUP	Power-on Counter	-	$2^{15}/f_{OSC2}$	-	s
tDVDD	Rising time of power line	-	-	3	ms
VPORH	Power-on Reset releasing voltage	2.8	3	3.2	V
VPORL	Power-on Reset detection voltage	2.6	2.8	3.0	V
tPORDT1	Power-on Reset release response time		30		μS

Note:Since the power-on releasing voltage and the power-on reset detection voltage relatively change, the detection voltage is never reversed.

Note:Due to the fluctuation of supply voltage, the power-on reset circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

20. Voltage Detection Circuit (VLTD)

The voltage detection circuit generates a reset signal by detecting a decreasing voltage.

Note: Due to the fluctuation of supply voltage, the voltage detection circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

20.1 Structure

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 (DVDD5B) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDLVL), and the comparator compares it with the reference voltage. When the supply voltage (DVDD5B) becomes lower than the detection voltage (VDLVL), a voltage detection reset signal is generated.

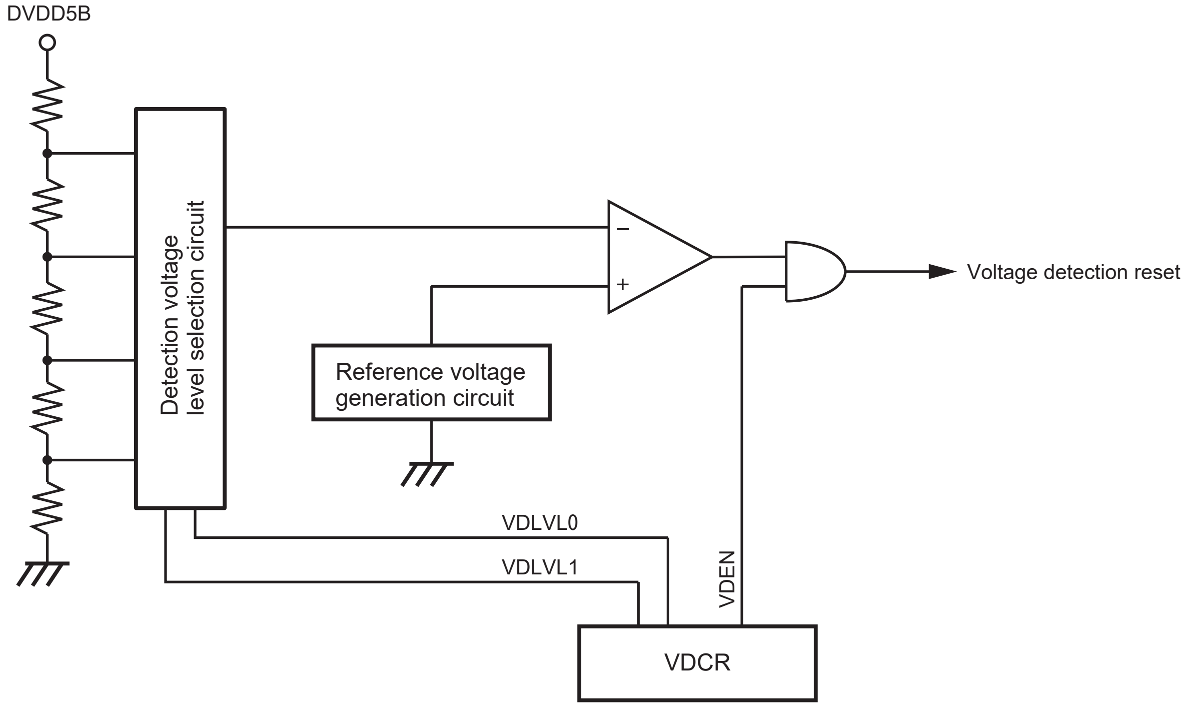

DVDD5B
Detection voltage level selection circuit
−
+
Voltage detection reset
Reference voltage generation circuit
VDLVL0
VDLVL1
VDEN
VDCR

Figure 20-1 Voltage Detection Circuit)

20.2 Registers

20.2.1 Register List

Base Address = 0x4004_0900

Register name		Address(Base+)
Voltage detection control register	VDCR	0x0000

20.2.2 VDCR (Voltage detection control register)

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

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-1	VDLVL[1:0]	R/W	Selection for detection voltage 00: Reserved 01: 4.1 ± 0.2V 10: 4.4 ± 0.2V 11: 4.6 ± 0.2V
0	VDEN	R/W	Voltage detection operation 0: Disabled 1: Enabled

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

20.3 Operation Description

20.3.1 Control

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

20.3.2 Function

The detection voltage can be selected by VDCR<VDLVL[1:0]>. Enabling/disabling the voltage detection can be programmed by VDCR<VDEN>. After the voltage detection operation is enabled, When the supply voltage (DVDD5B) becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection reset signal is generated.

20.3.2.1 Enabling/disabling the voltage detection operation

Setting VDCR<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation. VDCR<VDEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (DVDD5B) is lower than the detection voltage VDCR<VDLVL[1:0]>, setting VDCR<VDEN> to "1" generates reset signal at the time.

20.3.2.2 Selecting the detection voltage level

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

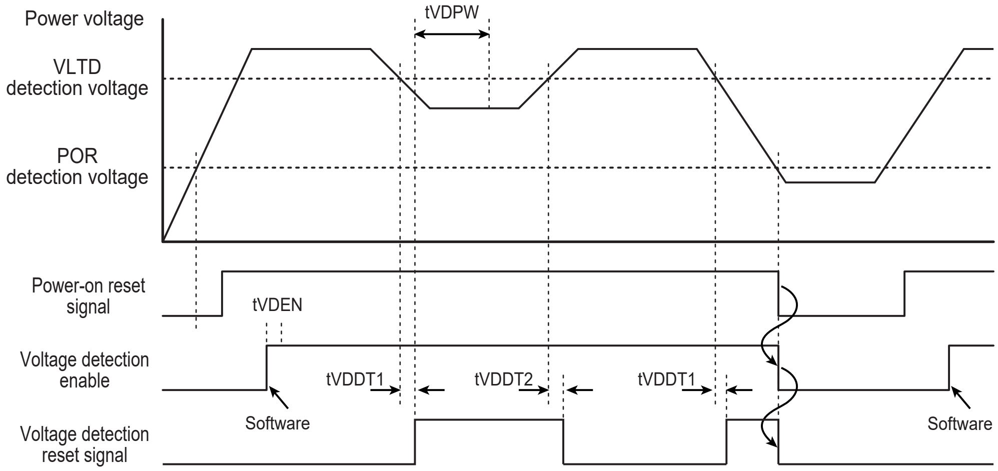


Figure 20-2 Voltage Detection Timing

Symbol	Parameter	Min	Typ	Max	Unit
tVDEN	Setup time after enabling voltage detection	-	40	-	μs
tVDDT1	Voltage detection response time	-	40	-	
tVDDT2	Voltage detection releasing time‘	-	40	-	
tVDPW	Voltage detection minimum pulse width	45	-	-	

21. Oscillation Frequency Detector (OFD)

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

The oscillation frequency detection is controlled by OFDCR1, OFDCR2 registers and the detection frequency range is specified by OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON which are the detection frequency setting registers. The lower detection frequency is specified by OFDMNPLLOFF/OFDMNPLLON registers and the higher detection frequency is specified by OFDMXPLLOFF/OFDMXPLLON registers.

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

When the TMPM37AFSQG detects the out of frequency by lower and higher detection frequency setting registers, all I/Os become high impedance by reset. In case of PLLOFF, OFDMNPLLOFF and OFDMXPLLOFF registers are valid for detection and the setting value of OFDMNPLLON/OFDMXPLLON registers are ignored. In case of PLLON, OFDMNPLLON and OFDMXPLLON registers are valid for detection and the setting value of OFDMNPLLOFF/OFDMXPLLOFF registers are ignored. By the oscillation frequency detection reset, all I/Os except power supply pins, $\overline{\text{RESET}}$ pin and MODE pin become high impedance and the CPU is also initialized. As the CG registers are initialized, so the PLL operation of the system clock is disabled and the clock operation is restarted after switching an oscillator to on chip oscillation (fosc2).

Since all registers for oscillation frequency detector (OFDCR1/OFDCR2/OFDMNPLLOFF/OFDMNPLLON/OFDMXPLLOFF/OFDMXPLLON) are not initialized by the reset generated from oscillation frequency detector.

And so when the oscillation frequency detection reset is generated, the system clock is switched to on chip oscillation and the reset sequence is executed just as the oscillation frequency detection is enabled.

Note: It is not guaranteed that OFD can detect all defects at any time, and it is not a circuit to measure error frequency.

Note: The oscillation frequency detection reset is available only in NORMAL and IDLE modes. In STOP mode, the oscillation frequency detection reset is disabled automatically.

Note: When the PLL is controlled (enabled or disabled) by the CGPLLSEL register or when the system clock is changed (fosc1 or fosc2) by the <OSCSEL> of CGOSCCR register, the OFD must be disabled beforehand. If OFD reset is generated with PLL-ON, the detection frequency setting registers (OFDMNPLLON/OFDMXPLLON) are automatically switched over to OFDMNPLLOFF/OFDMXPLLOFF.

21.1 Block diagram

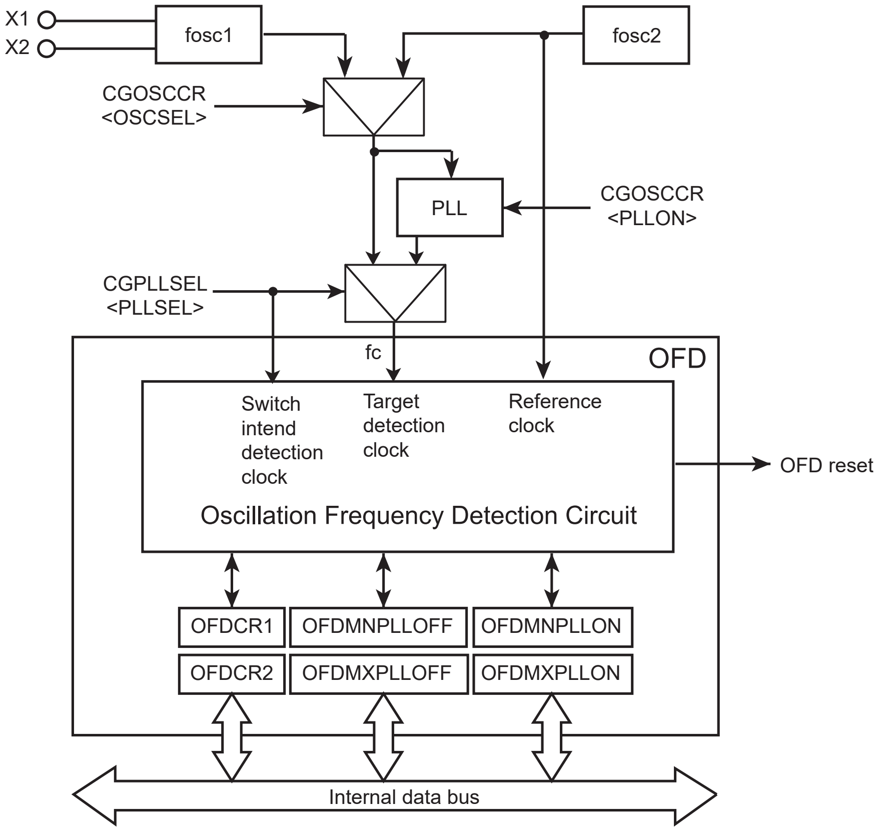


Figure 21-1 Oscillation Frequency Detector Block diagram

21.2 Registers

21.2.1 Register List

Base Address = 0x4004_0800

Register name		Address(Base+)
Oscillation frequency detection control register 1	OFDCR1	0x0000
Oscillation frequency detection control register 2	OFDCR2	0x0004
Lower detection frequency setting register (PLL OFF)	OFDMNPLLOFF	0x0008
Lower detection frequency setting register (PLL ON)	OFDMNPLLON	0x000C
Higher detection frequency setting register (PLL OFF)	OFDMXPLLOFF	0x0010
Higher detection frequency setting register (PLL ON)	OFDMXPLLON	0x0014

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

21.2.1.1 OFDCR1 (Oscillation frequency detection control register)

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

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDWEN[7:0]	R/W	Controls register write 0x06: Disable 0xF9: Enable Setting 0xF9 enables to write registers except OFDCR1. When writing a value except 0x06 or 0xF9, 0x06 is written. If writing register is disabled, reading from each register is enabled.

Note: OFDCR1 is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

21.2.1.2 OFDCR2 (Oscillation frequency detection control register 2)

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

Bit	Bit Symbol	Type	Description
31-8	−	R	Read as 0.
7-0	OFDEN[7:0]	R/W	Controls frequency detecting. 0x00: Disable 0xE4: Enable Writing a value except 0x00 or 0xE4 is invalid and a value will not be changed.

Note: OFDCR2 is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

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

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMNPLLOFF[8:0]	R/W	Sets internal lower detection frequency.

Note: Writing to the register of OFDMNPLLOFF is protected while OFD circuit is operating.

Note: OFDMNPLLOFF is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

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

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMNPLLON [8:0]	R/W	Sets external lower detection frequency.

Note: Writing to the register of OFDMNPLLON is protected while OFD circuit is operating.

Note: OFDMNPLLON is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

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

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMXPLL-OFF[8:0]	R/W	Sets internal higher detection frequency.

Note: Writing to the register of OFDMXPLLOFF is protected while OFD circuit is operating.

Note: OFDMXPLLOFF is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

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

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

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMXPLLON [8:0]	R/W	Sets external higher detection frequency.

Note: Writing to the register of OFDMXPLLON is protected while OFD circuit is operating.

Note: OFDMXPLLON is initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset.

21.3 Operational Description

21.3.1 Setting

Registers of OFD are initialized by the $\overline{\text{RESET}}$ pin, power on reset or VLTD reset. All register except OFDCR1 cannot be written by reset. They are able to be written by writing "0xF9" to OFDCR1.

The range of detection frequency is setting by OFDMNPLLON/OFDMXPLLON or OFDMNPLLOFF/OFDMXPLLOFF for each target clock. Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection.

To protect the mistaken writing, shout be written "0x06" to OFDCR1. And the register should be modified when OFD is stopped.

OFDMNPLLOFF/OFDMXPLLOFF and OFDMNPLLON/OFDMXPLLON are automatically switched over by the setting of CGPLLSEL<PLLSEL>.

When STOP mode is executed with OFDCR2=0xE4, the oscillation frequency detection is automatically disabled. After releasing STOP mode and warming up period, the oscillation frequency detection is enabled. The oscillation frequency detection is available only in NORMAL and IDLE mode. Table 21-1 shows the availability of oscillation frequency detector.

Table 21-1 Availability of oscillation frequency detector

Operating Mode	Oscillation Frequency Detection (OFDCR2=0xE4)	All I/Os condition after Oscillation Frequency Detection RESET (Except power supply, $\overline{\text{RESET}}$, 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	-

21.3.2 Operation

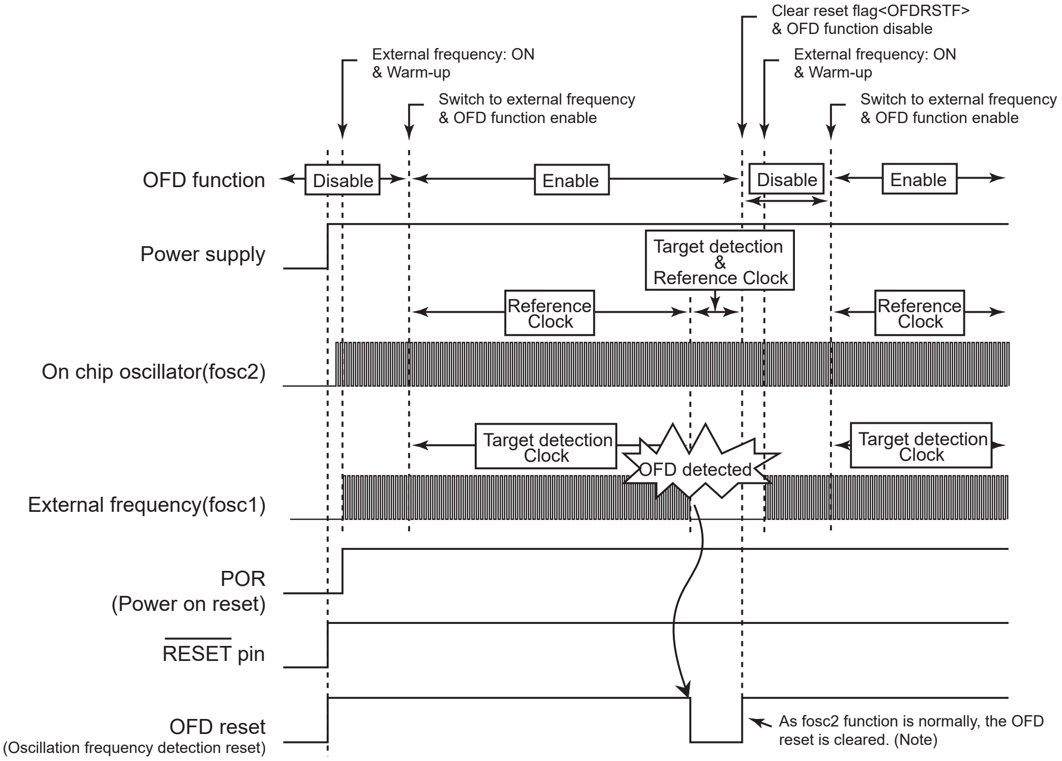
From the operation start-up to detection start-up, time length as two cycle of detecting clock is needed. And the Detecting cycle is 128/reference clock frequency.

When generating reset is enabled, OFD generates reset if the target clock frequency exceeds the frequency limit set by OFDMNPLLON/OFDMXPLLON and OFDMNPLLOFF/OFDMXPLLOFF. From detection of abnormal to reset generation, time length as one cycle of detecting clock is needed. The reset generated by OFD does not make itself and OFD continues detected operation.

Therefore, fosc is initialized to fosc2 and the target clock(fc) changes to fosc2 and detected operation is continued in PLL OFF.

Note: There are several factors of reset. Clock generator register CGRSTFLG can confirm the factors. For details of CGRSTFLG see chapter "exception."

Note: The target clock is set without 10MHz into OFDMNPLLOFF and OFDMXPLLOFF. (eg, setting to 8MHz value to these registers) And when the target clock frequency is fosc1 and the reset generated by OFD, the reset might be generated continuously due to as abnormal detection until OFD detects correctly.



Note: The target clock is set without 10MHz into OFDMNPLLOFF and OFDMXPLLOFF. (eg, setting to 8MHz value to these registers) And when the target clock frequency is fosc1 and the reset generated by OFD, the reset might be generated continuously due to as abnormal detection until OFD detects correctly.

Figure 21-2 Example of oscillation frequency detection operation

21.3.3 Detection Frequency

The detection frequency have a detection frequency range and an undetectable frequency range because of oscillation accuracy. Therefore, it is undefined whether to be detected between detection frequency range and undetectable it.

The upper and lower limit of detecting frequency are calculated by the maximum error of a target clock and a reference clock.

By the way of rounding the calculated result when OFDMNPLLON/OFDMNPLLOFF and OFDMXPLLON/ OFDMXPLLOFF are decided, the upper and lower limit of detecting and undetecting range shown as follows. The way of rounding is selected depending on the unevenness of the detected clock.

- In case of rounding up OFDMXPLLON/OFDMXPLLOFF and rounding down OFDMNPLLON/ OFDMNPLLOFF

  The target clock is higher than the upper limit of undetecting range and lower than the lower limit of undetecting range.
- In case of rounding down OFDMXPLLON/OFDMXPLLOFF and rounding down OFDMNPLLON/ OFDMNPLLOFF

  The target clock is lower than the upper limit of undetecting range and higher than the lower limit of undetecting range.

How to calculate the setup value of OFDMXPLLOFF/OFDMNPLLOFF is shown below when the target clock error is ±% (undetecting range) and the reference clock error is ± %. In this example, OFDMXPLLOFF is rounded up and OFDMNPLLOFF is rounded down.(From "a" to "h" corresponds to "Figure 21-3 Example of detection frequency range (in case of 10MHz)")

Figure 21-3 shows the detection or undetectable and detectable frequency range.

Figure 21-3 Example of detection frequency range (in case of 10MHz)

21.3.4 Available Operation Mode

The oscillation frequency detection is available only external oscillation frequency in NORMAL and IDLE mode. Before shifting to another mode or using on chip oscillation frequency, disable the oscillation frequency detection.

21.3.5 Example of Operational Procedure

The example of operational procedure is shown below.

After reset, confirms various reset factor by CGRSTFLG. If the reset factor is not by the oscillation frequency detect, enable external oscillation, set register to use OFD and enable operation.

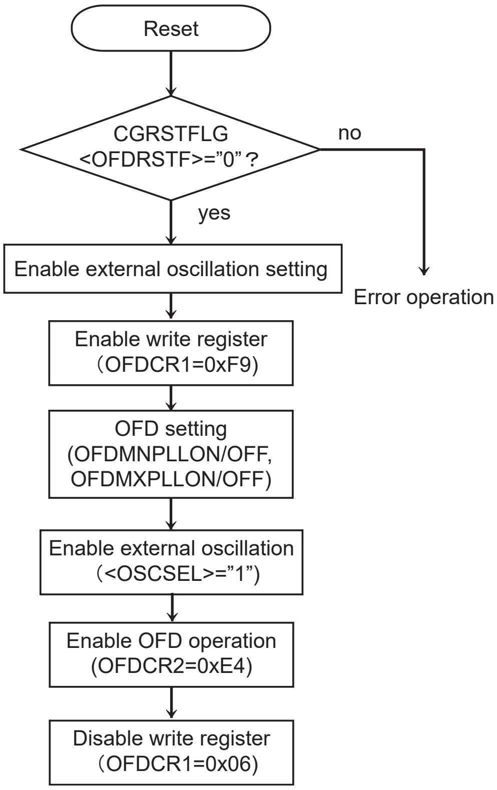


Figure 21-4 Example of operational procedure

22. 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 ($\overline{\text{WDTOUT}}$).

22.1 Configuration

Figure 22-1shows the block diagram of the watchdog timer.

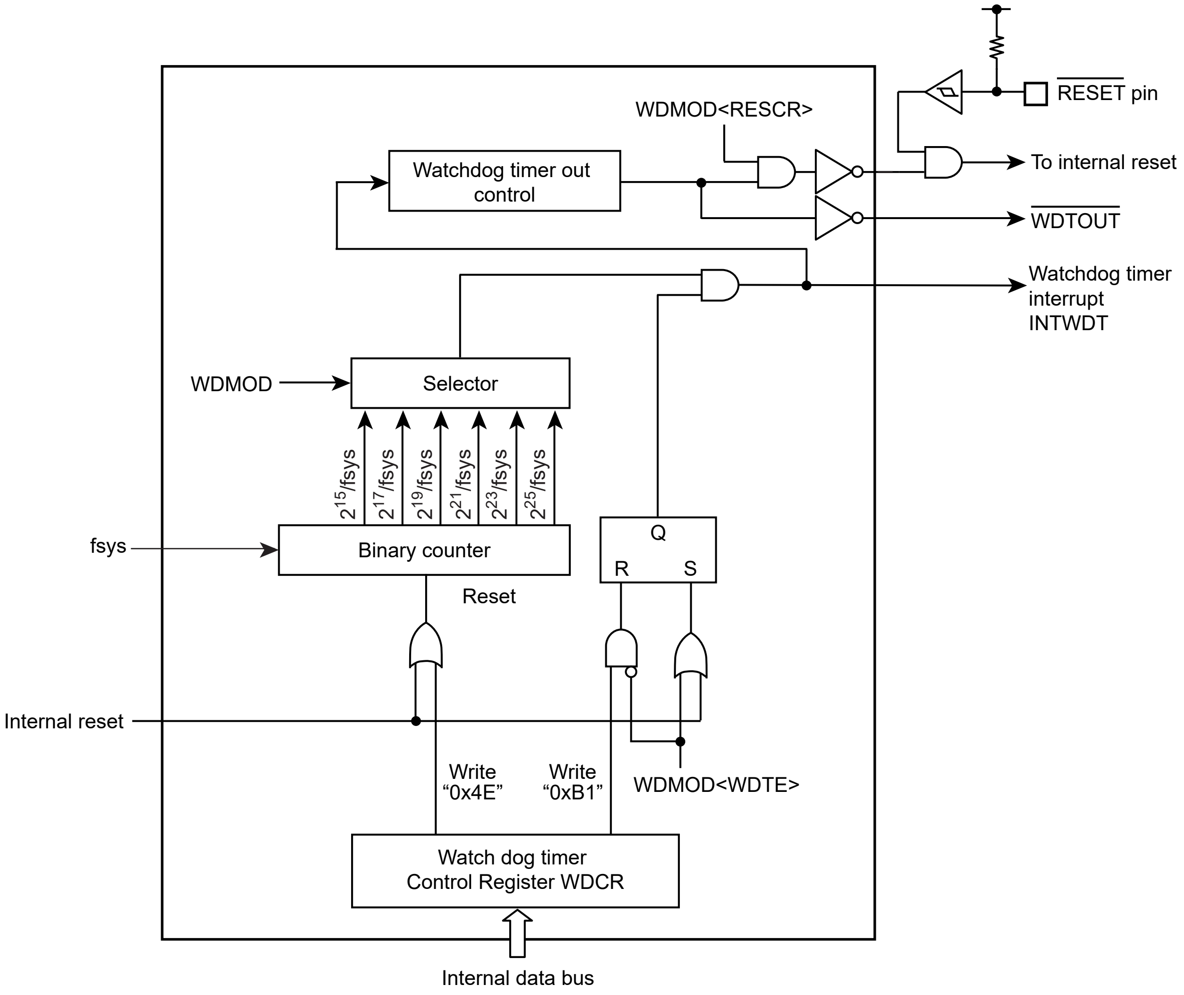


Figure 22-1 Block Diagram of the Watchdog Timer

22.2 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

22.2.1 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 22-1) 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 22-1 Detection time of watchdog timer (fc = 40MHz)

Clock gear value CGSYSCR<GEAR[2:0]>	WDMOD<WDTP[2:0]>					
	000	001	010	011	100	101
000 (fc)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
100 (fc/2)	1.63 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
101 (fc/4)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
110 (fc/8)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s
111 (fc/16)	13.12 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s	13.42 s

22.2.2 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

22.3 Operations

22.3.1 Basic Operation

The Watchdog timer is consists of the binary counters that work using the system clock (fsys) as an input. Detecting time can be selected between 2^{15}, 2^{17}, 2^{19} , 2^{21}, 2^{23} and 2^{25} by the WDMOD<WDTP[2:0]>. The detecting time as specified is elapsed, the watchdog timer interrupt (INTWDT) generates, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) output "Low".

To detect malfunctions (runaways) of the CPU caused by noise or other disturbances, the binary counter of the watchdog timer should be cleared by software instruction before INTWDT interrupt generates. If the binary counter is not cleared, the non-maskable interrupt generates by INTWDT. Thus CPU detects malfunction (runway), malfunction countermeasure program is performed to return to the normal operation.

Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

Note: This product does not include a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

22.3.2 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.

22.4 Operation when malfunction (runaway) is detected

22.4.1 INTWDT interrupt generation

In the Figure 22-2 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($\overline{\text{WDTOUT}}$).

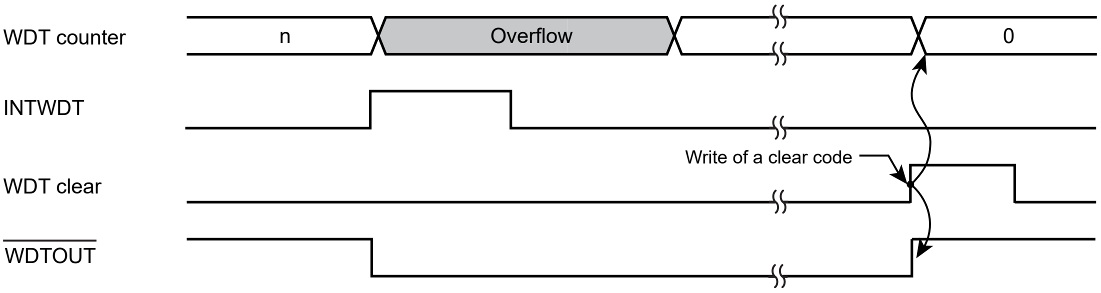


Figure 22-2 INTWDT interrupt generation

22.4.2 Internal reset generation

Figure 22-3 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.

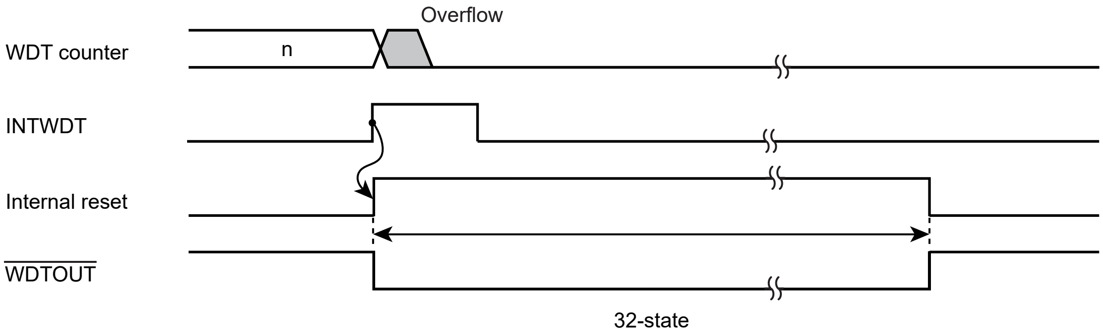


Figure 22-3 Internal reset generation

22.5 Control register

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

22.5.1 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.

22.5.2 Watchdog Timer Control Register(WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

22.5.3 Setting example

22.5.3.1 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).

22.5.3.2 Enabling control

Set WDMOD <WDTE> to "1".

		7	6	5	4	3	2	1	0	
WDMOD	←	1	-	-	-	-	-	-	-	Set <WDTE> to "1".

22.5.3.3 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).

22.5.3.4 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	-	-	-	-

23. Flash Memory Operation

This section describes the hardware configuration and operation of Flash memory. In this section, "1-word" means 32 bits.

23.1 Features

23.1.1 Memory Size and Configuration

Table 23-1 and Figure 23-1show a built-in memory size and configuration of TMPM37AFSQG.

Table 23-1 Memory size and configuration

Memory size	Block configuration				# of words per page	# of pages	Write time		Erase time	
	128 KB	64 KB	32 KB	16 KB			1 page	Total area	Block erase	Chip erase
64 KB	−	−	2	−	32	512	1.25ms	0.64 sec	0.1sec	0.2 sec

Note: The above values are theoretical values not including data transfer time. The write time per chip depends on the write method used by a user.

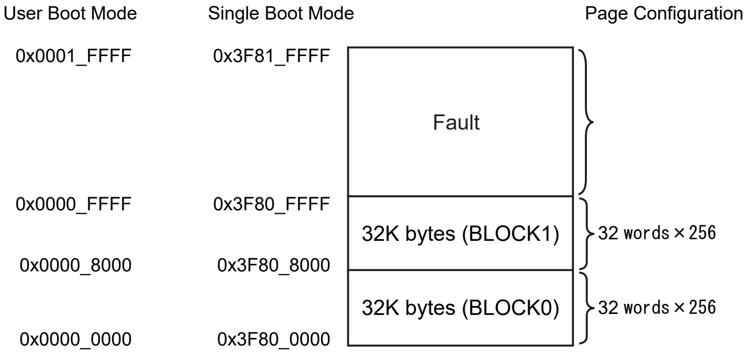


Figure 23-1 Block configuration

Flash memory configuration units are described as "block" and "page".

- Page

  One page is 32 words. Same address [31:7] is used in a page. First address of the group is [6:0] = 0 and the last address of the group is [6:0] = 0x7F.

- Block

  One block is 32KB and flash memory is consists of two blocks.

Write operation is performed per page. The write time per page is 1.25ms. (Typ.)

Erase is performed per block (auto block erase command use) or performed on entire flash memory (use of auto chip erase command). Erase time varies on commands. If auto block command is used, the erase time will be 0.1 sec per block (Typ.). If the auto chip erase command is used to erase entire area, the time will be 0.2 sec (Typ.).

In addition, the protect function can be used per block. For detail of the protect function, refer to "23.1.5 Protect/Security Function".

23.1.2 Function

Flash memory built-in this device is generally compliant with the JEDEC standards except for some specific functions. Therefore, if a user is currently using a flash memory as an external memory, it is easy to implement the functions into this device. Furthermore, to provide easy write or erase operation, this product contains a dedicated circuit to perform write or chip erase automatically.

JEDEC compliant functions	Modified, added, or deleted functions
・Automatic programming ・Automatic chip erase ・Automatic block erase ・Data polling/toggle bit	<Modified> Block write/erase protect (only software protection is supported) <Deleted> Erase resume - suspend function

23.1.3 Operation Mode

23.1.3.1 Mode Description

This device provides the single chip mode and single boot mode. The single chip mode contains the normal mode and user boot mode. Figure 23-2 shows the mode transition.

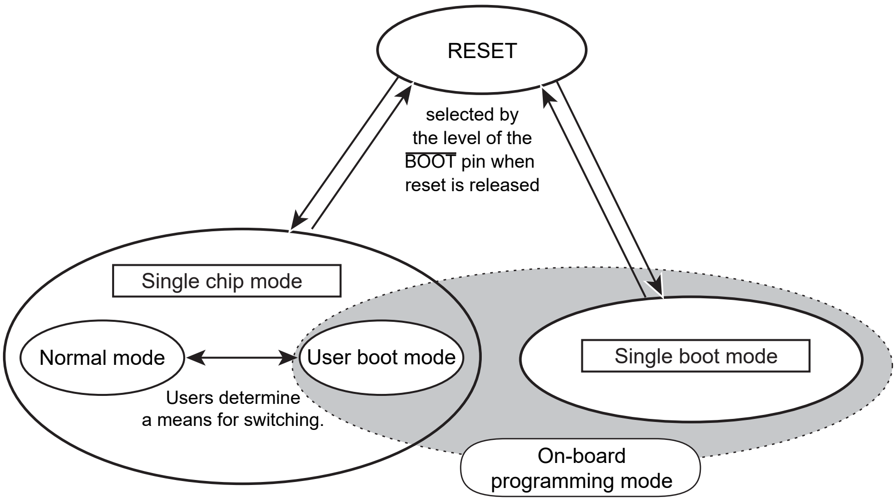

RESET
selected by the level of the $\overline{\text{BOOT}}$ pin when reset is released
Single chip mode
Normal mode
User boot mode
Users determine a means for switching.
Single boot mode
On-board programming mode

Figure 23-2 Mode transition

(1) Single chip mode

The single chip mode is a mode where the device can boot-up from Flash memory after reset. The mode contains two sub-modes in below.

- Normal mode

  The mode where user application program is executed.

- User boot mode

  The mode where flash memory is re-programmed on the user's set.

Users can switch the normal mode to user boot mode freely. For example, a user can set if PA0 of port A is "1", the mode is the normal mode. If PA0 of port A is "0", the mode is the user boot mode. The user must prepare a routine program in the application program to determine the switching.

(2) Single boot mode

The mode where flash memory can boot-up from the built-in BOOT ROM (Mask ROM) after reset.

The BOOT ROM contains the algorithm that can rewrite Flash memory via serial port of this device on the user's set. With connecting the serial port to external host, data transfer is performed in above-mentioned protocol and re-programed Flash memory.

(3) On-board programming mode

The user boot mode and single boot mode are the modes where flash memory can be re-programmable on the user's set. These two modes are called "on-board programming mode".

23.1.3.2 Mode Determination

Either the single chip or single boot operation mode can be selected by the level of the $\overline{\text{BOOT}}$ pin when reset is released.

Table 23-2 Operation mode setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

23.1.4 Memory Map

Figure 23-3shows a comparison of the memory map in the single chip mode and single boot mode. In the single boot mode, built-in Flash memory is mapped to 0x3F80_0000 and subsequent addresses, and the built-in BOOT ROM is mapped to 0x0000_0000 through 0x0000_0FFF.

Flash memory and RAM addresses are shown below.

FLASH size	RAM size	FLASH address	RAM address
64KB	4KB	0x0000_0000 to 0x0000_FFFF(single chip mode) 0x3F80_0000 to 0x3F80_FFFF(single boot mode)	0x2000_0000 to 0x2000_0FFF

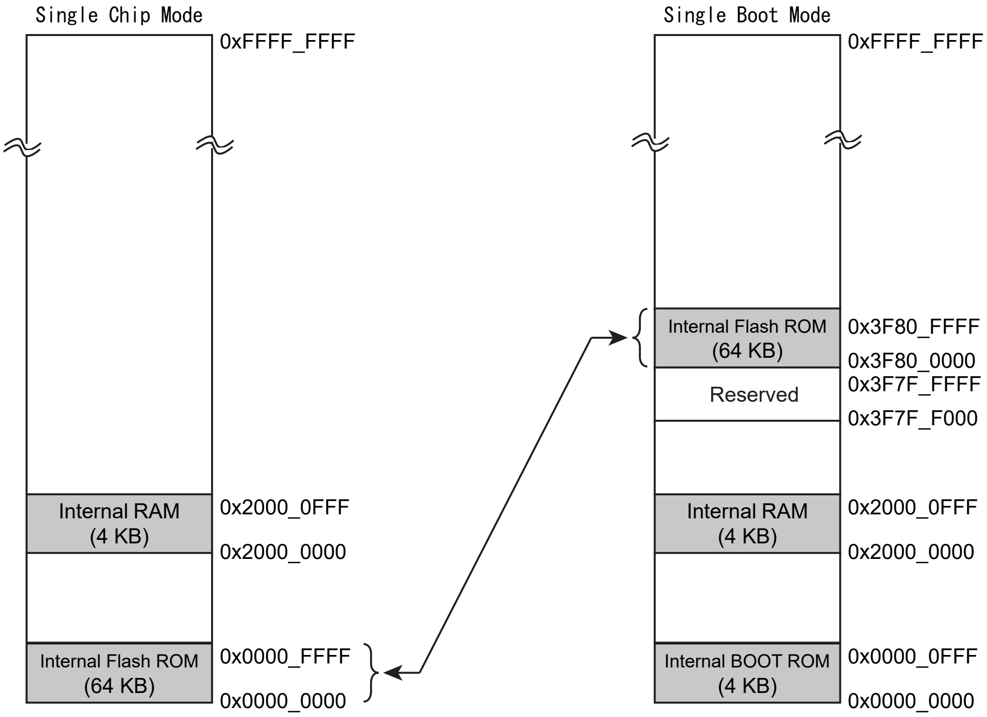


Figure 23-3 Comparison of memory map

23.1.5 Protect/Security Function

This device has the protect and security functions for Flash memory.

1. Protect function

   The write/erase operation can be inhibited per block.

2. Security function

   The read operation from a flash writer can be inhibited.

   Usage restrictions on debug functions

23.1.5.1 Protect Function

This function inhibits the write/erase operation per block.

To enable the protect function, a protect bit corresponding to a block is set to "1" using the protect bit program command. If a protect bit is set to "0" using the protect bit erase command, a block protect can be canceled. The protect bit can be monitored with FCPSRA<BLK[1:0]>.

A program of protect bit can be programmed by 1-bit unit and can be erased by 4-bit unit. For detail of programming/erasing of protect bits, refer to "23.2.5 Command Description".

23.1.5.2 Security Function

Table 23-3 shows operations when the security function is enabled.

Table 23-3 Operations when the security function is enabled.

Item	Description
Read flash memory	CPU can read flash memory.
Debug port	JTAG, serial wire or trace communication is disabled.
Command execution to Flash memory	Command write to flash memory is not accepted. If a user tries to erase a protect bit, chip erase is executed and all protect bits are erased.

The security function is enabled under the following conditions;

1. FCSECBIT<SECBIT> is set to "1".
2. All protect bits (FCPSRA<BLK>) are set to "1".

FCSECBIT<SECBIT> is set to "1" by the Cold Reset(Power OnReset) . Rewriting of FCSECBIT <SECBIT> is described in below.

Note: Use a 32-bit transfer instruction when the following writing operations, item1 and 2.

1. Write the specified code (0xa74a9d23) to FCSECBIT
2. Write data within 16 clocks after the operation of item 1.

23.1.6 Register

23.1.6.1 Register List

Base Address = 0x41FF_F000

Register name		Address(Base+)
Security bit register	FCSECBIT	0x0010
Flash Interface controlregister	FCCR	0x001C
Flash status register	FCSR	0x0020
Flash protect status register A	FCPSRA	0x0030

23.1.6.2 FCCR (Flash Interface control register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	FLBOFF	R/W	Control of Flash Interface with instruction Buffer (Note 1) 0: Enable Instruction Buffer 1: Disable Instruction Buffer (with Buffer clear) This bit is a functional bit for controlling the Flash Interface . To use Instruction Buffer, set "0". To not use Instruction Buffer ,set to "1". In TMPM37AFSQG, it must be set "0" for Flash accessing.

Note 1: In TMPM37AFSQG, after Flash programing or Flash Erasing ,it should be Clearing Instruction buffer by this functional bit or insert a reset signal .

Instruction Buffer Clearing operation as following.

after executing FCCR<FLBOFF>="1" , set FCCR<FLBOFF>="0" again on RAM.

23.1.6.3 FCSR (Flash status register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	RDY/BSY	R	Ready/Busy (Note 1) 0: Busy (during auto operation) 1: Ready (auto operation ends) This bit is a function bit to monitor flash memory from CPU. While flash memory is in auto operation, this bit outputs "0" to indicate that flash memory is busy. Once auto operation is finished, this bit becomes ready state and outputs "1". Then next command is accepted. If a result of auto operation is failed, this bit outputs "0" continuously. The bit returns to "1" by hardware reset.

Note 1: Make sure that flash memory is ready before commands are issued. If a command is issued during busy, not only the command is not sent but also subsequent commands may not be accepted. In that case, use hardware reset to return. Hardware reset needs 0.5 μs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset.

23.1.6.4 FCSECBIT (Security bit register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SECBIT	R/W	Security bit 0: Security function setting is disabled. 1: Security function setting is enabled.

Note: This register is initialized by Cold Reset(Power OnReset) .

23.1.6.5 FCPSRA (Flash protect status register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	BLK1	BLK0
After reset	0	0	0	0	0	0	(Note 1)	(Note 1)

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1-0	BLK1- BLK0	R	Protection status of Block1 to 0 0: Not protected 1: Protected Protect bit values correspond to protect status of each block. If corresponding bit indicates "1", corresponding block is in the protection status. A block in the protection status cannot be re-programmable.

Note 1: A value will correspond to the protection status.

23.2 Detail of Flash Memory

In on-board programming, the CPU executes commands for reprogramming or erasing Flash memory. This reprogramming/erase control program should be prepared by the user beforehand. Since Flash memory content cannot be read while Flash memory is being written or erased, it is necessary to run the reprogram/erase control program on the built-in RAM. Do not generate interrupt/fault except reset to avoid abnormal program termination.

23.2.1 Function

Flash memory is generally compliant with the JEDEC standards except for some specific functions. However; a method of address designation of operation command is different from standard commands.

If write/erase operation is executed, commands are input to flash memory using 32-bit (1-word) store instruction command. After command input, write or erase operation is automatically executed in inside.

Table 23-4 Flash memory function

Main function	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of Flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Write/erase protect	The write or erase operation can be individually inhibited for each block.

Note: In TMPM37AFSQG, after Flash programing or Flash Erasing ,it should be Clearing Instruction buffer. Pls refter "23.1.6.2 FCCR (Flash Interface control register)" for a clear method.

Note: Check the FCSR<RDY/BSY> to make sure each command sequence end such as Flash writing,Flash Erase, Protection bit program, Protection bit Erase . and then hold for 200 μs or more before reading data from Flash memory or starting instruction fetch.

23.2.2 Operation Mode of Flash Memory

Flash memory provides mainly two types of operation modes;

- The mode to read memory data (Read mode)
- The mode to erase or rewrite memory data automatically (Automatic operation mode)

After power-on, after rest or after automatic operation mode is finished normally, Flash memory becomes read mode. Instruction stored in Flash memory or data read is executed in the read mode.

If commands is input during the read mode, the operation mode becomes the automatic operation. If the command process is normally finished, the operation mode returns to the read mode except the ID-Read command. During the automatic operation, data read and instruction execution stored in Flash memory cannot be performed.

If command process is abnormally finished then the operation mode should forcibly return to read mode. In this case, use the read command, read/reset command or hardware reset.

23.2.3 Hardware Reset

A hardware reset means a Cold Reset(Power OnReset) or warm reset to use returning to the read mode when the automatic programming/erase operation is forcibly cancelled, or automatic operation abnormally ends.

If the hardware reset occurs during the automatic operation, Flash memory stops the automatic operation and returns to the read mode. If a hardware reset is generated during Flash memory automatic program/erase operation, the hardware reset needs 0.5 μs or more reset period regardless of system clock. At this time, it takes approximately 2 ms until enabling to read after reset . Note that if a hardware reset occurs during the automatic operation, data write operation is not executed properly. Set write operation again.

For detail of the reset operation, refer to "Reset". After a given reset input, CPU will read the reset vector data and then starts the routine after reset.

23.2.4 How to Execute Command

The command execution is performed by writing command sequences to Flash memory with a store instruction. Flash memory executes each automatic operation command according to the combination of input addresses and data. For detail of the command execution, refer to "23.2.5 Command Description".

An execution of store instruction to the Flash memory is called "bus write cycle". Each command consists of some bus write cycles. In Flash memory, when address and data of bus write cycle are performed in the specified order, the automatic command operation is performed. When the cycle is performed in non-specified order, Flash memory stops command execution and returns to the read mode.

If you cancel the command during the command sequence or input a different command sequence, execute the read command or read/reset command. Then Flash memory stops command execution and returns to the read mode. The read command and read/reset command are called "software reset".

When write command sequence ends, the automatic operation starts and FCSR<RDY/BSY> is set to "0". When the automatic operation normally ends, FCSR<RDY/BSY> = "1" is set and Flash memory returns to the read mode.

New command sequences are not accepted during the automatic operation. If you want to stop the command operation, use a hardware reset. In case that the automatic operation abnormally ends (FCSR<RDY/BSY> remains "0"), Flash memory remains locked and will not return to the read mode. To returns to the read mode, use a hardware reset. If the hardware reset stops the command operation, commands are not normally executed.

Notes on the command execution;

1. To recognize command, command sequencer need to be in the read mode before command starting. Confirm FCSR<RDY/BSY> = 1 is set prior to the first bus write cycle of each command. Consecutively, it is recommended that the read command is executed.

2. Execute each command sequence from outside of Flash memory.

3. Execute sequentially each bus write cycle by data transfer instruction in one-word (32-bit).

4. Do not access Flash memory during the each command sequence. Do not generate any interrupt or fault except reset.

5. Upon issuing a command, if any address or data is incorrectly written, make sure to return to the read mode by using software reset.

23.2.5 Command Description

This section explains each command content. For detail of specific command sequences, refer to "23.2.6 Command Sequence".

23.2.5.1 Automatic Page Program

(1) Operation Description

The automatic page program writes data per page. When the program writes data to multiple pages, a page command need to be executed in page by page. Writing across pages is not possible.

Writing to Flash memory means that data cell of "1" becomes data of "0". It is not possible to become data cell of "1" from data of "0". To become data cell of "1" from "0", the erase operation is required.

The automatic page program is allowed only once to each page already erased. Either data cell of "1" or "0" cannot be written data twice or more. If rewriting to a page that has already been written once, the automatic page program is needed to be set again after the automatic block erase or automatic chip erase command is executed.

Note 1: Page program execution to the same page twice or more without erasing operation may damage the device.

Note 2: Writing to the protected block is not possible.

(2) How to Set

The 1st to 3rd bus write cycles indicate the automatic page program command.

In the 4th bus write cycle, the first address and data of the page are written. On and after 5th bus cycle, one page data will be written sequentially. Data is written in one-word unit (32-bit).

If a part of the page is written, set "0xFFFFFFFF" as data, which means not required to write, for entire one page.

No automatic verify operation is performed internally in the device. So, be sure to read the data programmed to confirm that it has been correctly written.

If the automatic page program is abnormally terminated, that page has been failed to write. It is recommended not to use the device or not to use the block including the failed address.

23.2.5.2 Automatic Chip Erase

(1) Operation Description

The automatic chip erase is executed to the memory cell of all addresses. If protected blocks are contained, these blocks will not be erased. If all blocks are protected, the automatic chip erase operation will not performed and will return to the read mode after a command sequence is input.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic chip erase command. After the command sequence is input, the automatic chip erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

23.2.5.3 Automatic Block Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified block. If the specified block is protected, erase operation is not executed.

(2) How to Set

The 1st to 5th bus write cycles indicate the automatic block erase command. In the 6th bus write cycle, the block to be erased is specified. After the command sequence is input, the automatic block erase operation starts.

No automatic verify operation is performed internally in the device. So, be sure to read the data to confirm that it has been correctly erased.

23.2.5.4 Automatic Protect Bit Program

(1) Operation Description

The automatic protect bit program writes "1" to a protect bit at a time. To set "0" to a protect bit, use the automatic protect bit erase command.

For detail of the protect function, refer to "23.1.5 Protect/Security Function".

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit program command. In the 7th bus write cycle, the protect bit to be written is specified. After the command sequence is input, the automatic protect bit program starts. Check whether write operation is normally terminated with FCPSRA<BLK>.

23.2.5.5 Auto Protect Bit Erase

(1) Operation Description

The automatic protect bit erase command operation depends on the security status. For detail of security status, refer to "23.1.5 Protect/Security Function".

- Non-security status

  Clear the specified protect bit to "0". Protect bit erase is performed in 4-bit unit.
- Security status

  Erase all protect bits after all addresses of Flash memory are erased.

(2) How to Set

The 1st to 6th bus write cycles indicate the automatic protect bit erase command. In the 7th bus write cycle, the protect bit to be erased is specified. After the command sequence is input, the automatic protect bit erase operation starts.

In the non-security status, specified protect bit is erased. Check whether erase operation is normally terminated with FCPSRA<BLK>.

In the security status, all addresses and all protect of Flash memory bits are erased. Confirm if data and protect bits are erased normally. If necessary, execute the automatic protect bit erase, automatic chip erase or automatic block erase.

All cases are the same as other commands, FCSR<RDY/BSY> becomes "0" during the automatic protect bit erase command operation. After the operation is complete, FCSR<RDY/BSY> becomes "1" and Flash memory will return to the read mode. To abort the operation, a hardware reset is required.

23.2.5.6 ID-Read

(1) Operation Description

The ID-Read command can read information including Flash memory type and three types of codes such as a maker code, device code and macro code.

(2) How to Set

The 1st to 3rd bus write cycles indicate the ID-Read command. In the 4th bus write cycle, the code to be read is specified. After the 4th bus write cycle, read operation in the arbitrary flash area acquires codes.

The ID-Read can be executed successively. The 4th bus write cycle and reading ID value can be executed repeatedly.

The ID-Read command does not automatically return to the read mode. To return to the read mode, execute the read command, read/reset command or hardware reset.

23.2.5.7 Read Command and Read/reset Command (Software Reset)

(1) Operation Description

A command to return Flash memory to the read mode.

When the ID-Read command is executed, macro stops at the current status without automatically return to the read mode. To return to the read mode from this situation, use the read command or read/reset command. It is also used to cancel the command when commands are input to the middle.

(2) How to Set

The 1st bus cycle indicates the read command. The 1st to 3rd bus write cycles indicate the read/reset command. After either command sequence is executed, Flash memory returns to the read mode.

23.2.6 Command Sequence

23.2.6.1 Command Sequence List

Table 23-5 shows addresses and data of bus write cycle in each command.

All command cycles except the 5th bus cycle of ID-Read command are bus write cycles. A bus write cycle is performed by 32-bit (1-word) data transfer instruction. (Following table shows only lower 8 bits of data.)

For detail of addresses, refer to Table 23-6. Use below values to "command" described in a column of Addr[15:9] in the Table 23-6.

Note 1) Always set to "0" to the address bit [1:0].

Note 2) Set below values to the address bit [19] according to Flash memory size.
Memory size is 1MB or less : Always set to "0"
Memory size is over 1MB : If bus write to 1MB area or less, the bit is set to "0".
If bus write to over 1MB area, the bit is set to "1".

Table 23-5 Command Sequence

Command	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	−	−	−	−	−	−
	0xF0	−	−	−	−	−	−
Read/reset	0xX55X	0xXAAX	0xX55X	−	−	−	−
	0xAA	0x55	0xF0	−	−	−	−
ID-Read	0xX55X	0xXAAX	0xX55X	IA	0xXX	−	−
	0xAA	0x55	0x90	0x00	ID	−	−
Automatic page program	0xX55X	0xXAAX	0xX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	−
	0xAA	0x55	0x80	0xAA	0x55	0x10	−
Automatic block erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	BA	−
	0xAA	0x55	0x80	0xAA	0x55	0x30	−
Automatic protect bit program	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Automatic protect bit erase	0xX55X	0xXAAX	0xX55X	0xX55X	0xXAAX	0xX55X	0xXX
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- IA: IDAddress
- ID: ID data
- PA: Program page address
- PD: Program data (32-bit data)

  After the 4th bus cycle, input data in the order of the addresses per page
- BA: Block address (see Table 23-7)

- PBA: Protect bit address (see Table 23-8)

23.2.6.2 Address Bit Configuration in the Bus Cycle

Table 23-6 is used in conjunction with "Table 23-5 Command Sequence".

Set the address setting according to the normal bus write cycle address configuration from the first bus cycle.

Table 23-6 Address bit configuration in the bus write cycle

<table>
<thead>
<tr><th>Address</th><th>Addr
[31:15]</th><th>Addr
[14]</th><th>Addr
[13:12]</th><th>Addr
[11:9]</th><th>Addr
[8:7]</th><th>Addr
[6:4]</th><th>Addr
[3:0]</th></tr>
</thead>
<tbody>
<tr><td rowspan="2">Normal Command</td><td colspan="7">Normal bus write cycle address configuration</td></tr>
<tr><td>Flash area</td><td colspan="2">"0" is recommended.</td><td>Command</td><td colspan="3">Addr[1:0] = "0" (fixed)
Other bits = "0" (recommended)</td></tr>
<tr><td rowspan="2">ID-READ</td><td colspan="7">IA: ID address (Setting of the 4th bus write cycle address for ID-READ)</td></tr>
<tr><td>Flash area</td><td>"0" is recommended.</td><td>ID Address</td><td colspan="4">Addr[1:0] = "0" (fixed)
Other bits= "0" (recommended)</td></tr>
<tr><td rowspan="2">Block erase</td><td colspan="7">BA: Block address(Setting of the 6th bus write cycle address for block erase)</td></tr>
<tr><td>Block address (Table 23-7)</td><td colspan="6">Addr[1:0] = "0" (fixed) Other bits= "0" (recommended)</td></tr>
<tr><td rowspan="2">Automatic page program</td><td colspan="7">PA: Program page address (Setting of the 4th bus write cycle address for page program)</td></tr>
<tr><td colspan="5">Page address</td><td colspan="2">Addr[1:0] = "0" (fixed)
Other bits= "0" (recommended)</td></tr>
<tr><td rowspan="2">Protect bit program</td><td colspan="7">PBA: Protect bit address (Setting of the 7th bus write cycle address for protect bit program)</td></tr>
<tr><td>Flash area</td><td colspan="3">Fix to "0"</td><td>Protect bit selection
(Table 23-8)</td><td colspan="2">Addr[1:0] = "0" (fixed)
Other bits= "0" (recommended)</td></tr>
</tbody>
</table>

23.2.6.3 Block Address(BA)

Table 23-7 shows block addresses. Specify any address included in the block to be erased in the 6th bus write cycle of the automatic block erase command.

Table 23-7 Block address

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
1	0x0000_8000 to 0x0000_FFFF	0x3F80_8000 to 0x3F80_FFFF	32
0	0x0000_0000 to 0x0000_7FFF	0x3F80_0000 to 0x3F80_7FFF	32

23.2.6.4 How to Specify Protect Bit (PBA)

The protect bit is specified in 1-bit unit in programming and in 4-bit unit in erasing.

Table 23-8 shows a protect bit selection table of the automatic protect bit program. The column of address example indicates an address described in upper side is used in the use boot mode and the lower side is used in the single boot mode.

Four protect bits are erased by the automatic protect bit erase command in all.

Table 23-8 Protect bit program address

Block	Protect bit	Address of 7th bus write cycle			Address example [31:0]
		Address [14:9]	Address [8]	Address [7]	
Block0	<BLK[0]>	Fix to "0"	0	0	0x0000_0000 0x3F80_0000
Block1	<BLK[1]>		0	1	0x0000_0080 0x3F80_0080

23.2.6.5 ID-Read Code (IA, ID)

Table 23-9 shows how to specify a code and the content using ID-Read command.

The column of address example indicates an address described in the upper side is used in the use boot mode and the lower side is used in the single boot mode

Table 23-9 ID-Read Command codes and contents

Code	ID[7:0]	IA[13:12]	Address Example [31:0]
Manufacture code	0x98	0b00	0x0000_0000 0x3F80_0000
Device code	0x5A	0b01	0x0000_1000 0x3F80_1000
−	Reserved	0b10	−
Macro code	0x33	0b11	0x0000_3000 0x3F80_3000

23.2.6.6 Example of Command Sequence

(1) use boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x0000_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	–	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_00F0	–	–	–	–
ID-Read	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550	0x0000_0AA0	0x0000_0550	0x0000_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

(2) Data single boot mode

Command		Bus cycle						
		1	2	3	4	5	6	7
Read	Address	0x3F80_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
Read/reset	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–	–	–	–
	Data	0x0000_00AA	0x3F80_0055	0x3F80_00F0	–	–	–	–
ID-Read	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	IA	0x0000_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Automatic page program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PA	In the following cycles, write addresses and data successively per page.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic chip erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Automatic block erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic protect bit program	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	PBA
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A
Automatic protect bit erase	Address	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550	0x3F80_0AA0	0x3F80_0550	0x3F80_0550
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A

23.2.7 Flowchart

23.2.7.1 Automatic Program

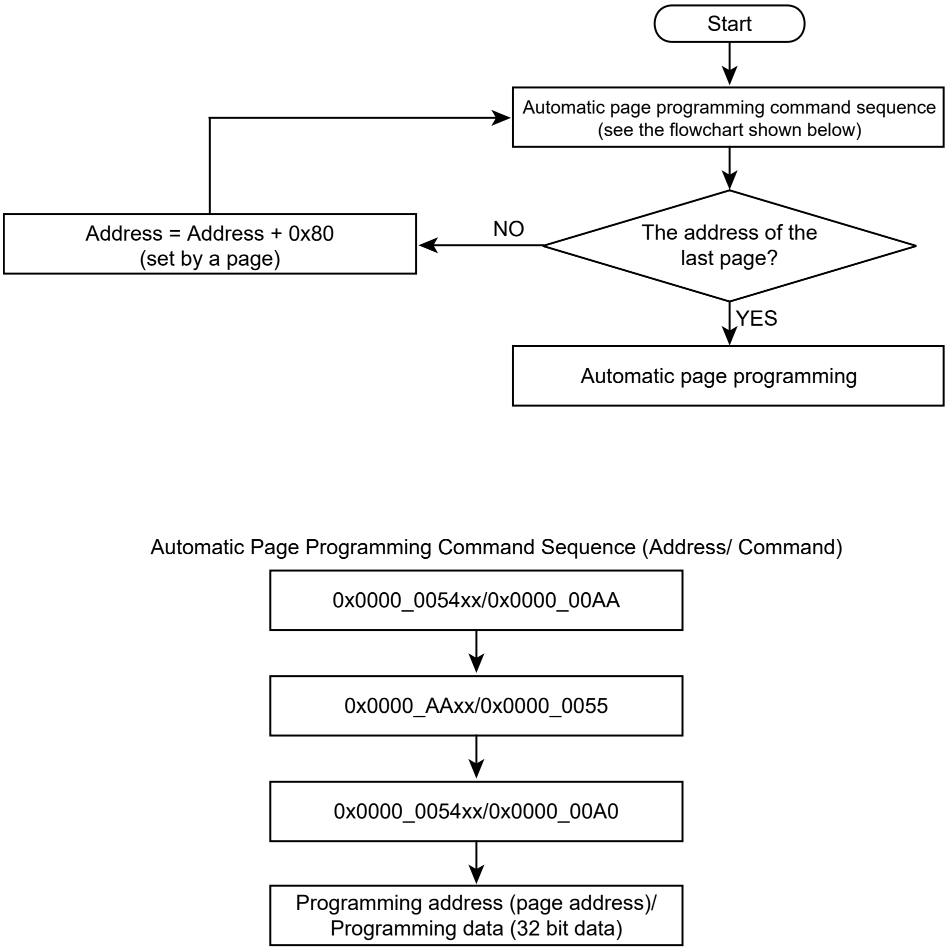

Start
Automatic page programming command sequence (see the flowchart shown below)
The address of the last page?
NO
Address = Address + 0x80 (set by a page)
YES
Automatic page programming
Automatic Page Programming Command Sequence (Address/ Command)
0x0000_0054xx/0x0000_00AA
0x0000_AAxx/0x0000_0055
0x0000_0054xx/0x0000_00A0
Programming address (page address)/ Programming data (32 bit data)

Figure 23-4 Flowchart of automatic program

23.2.7.2 Automatic Erase

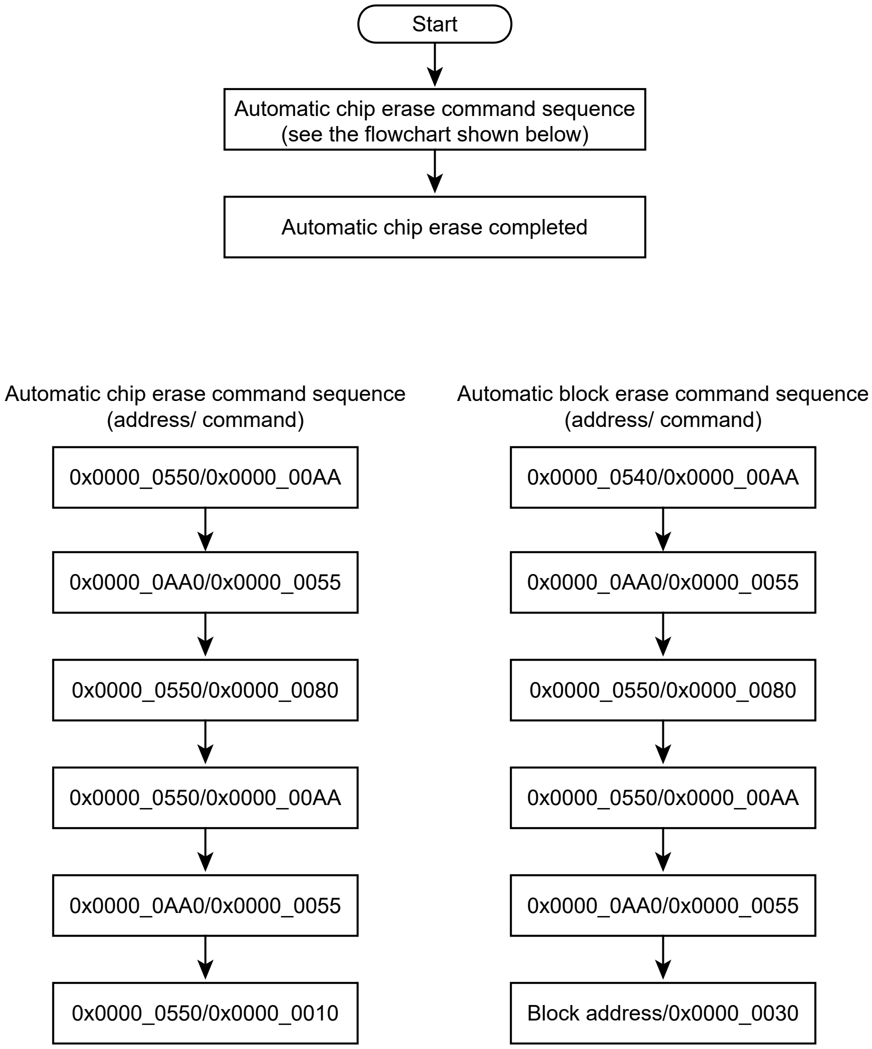


Figure 23-5 Flowchart of automatic erase

23.3 How to Reprogram Flash using Single Boot Mode

The single boot mode utilizes a program contained in built-in BOOT ROM for reprogramming Flash memory. In this mode, BOOT ROM is mapped to the area containing interrupt vector tables and Flash memory is mapped to another address area other than BOOT ROM area.

In the boot mode, Flash memory is reprogrammed using serial command/data transfer. With connecting serial channel (UART) of this device to the external host, a reprogramming program is copied from the external host to the built-in RAM. A reprogramming routine in the RAM is executed to reprogram Flash memory. For details of communication with host, follow the protocol described later.

Even in the single boot mode, do not generate interrupt/fault except reset to avoid abnormal program termination.

To secure the contents of Flash memory in the single chip mode (normal operation mode), once re-programming is complete, it is recommended to protect relevant flash blocks against accidental erasure during subsequent single chip operations.

23.3.1 Mode Setting

In order to execute the on-board programming, this device is booted-up in the single boot mode. Below setting is for the single boot mode setting.

$\overline{\text{BOOT}}$ = 0
$\overline{\text{RESET}}$ = 0 → 1

While $\overline{\text{BOOT}}$ pin is set to the above in advance, set $\overline{\text{RESET}}$ pin to "0". Then release $\overline{\text{RESET}}$ pin, the device will boot-up in the single boot mode.

23.3.2 Interface Specification

This section describes UART communication format in the single boot mode. The serial operation supports UART (asynchronous communication) modes. In order to execute the on-board programming, set the communication format of the programming controller as well.

- UART communication

  Communication channel: channel 0

  Serial transfer mode: UART (asynchronous), half-duplex, LSB first

  Data length: 8-bit

  Parity bit: None

  STOP bit: 1-bit

  Baud rate: Arbitrary baud rate

The boot program operates the clock/mode control block setting as an initial condition. For detail of the initial setting of the clock, refer to "Clock/Mode control".

As explained in the "23.3.5.1 Serial Operation Mode Determination", a baud rate is determined by the 16-bit timer (TMRB). When determining the baud rate, communication is executed by 1/16 of a desired baud rate. Therefore, the communication baud rate must be within the measurable range. The timer count clock operates at ΦT1 (fc/2).

A handshaking pin of I/O interface mode outputs "Low" waiting in receive state and outputs "High" in transmission state. Check the handshaking pin before communications and must follow the communication protocol.

Table 23-10 shows the pins used in the boot program. Other than these pins are not used by the boot program.

Table 23-10 Pin connection

Pin		Interface
		UART
Mode setting pin	$\overline{\text{BOOT}}$	o
Reset pin	$\overline{\text{RESET}}$	o
Communication pin	TXD0 (PE0)	o
	RXD0 (PE1)	o

o:used ×:unused

23.3.3 Restrictions on Internal Memories

Note that the single boot mode places restrictions on the built-in RAM and built-in flash memory as shown in Table 23-11.

Table 23-11 Restrictions on the memories in the single boot mode

Memory	Restrictions
Internal RAM	Boot program uses the memory as a work area through 0x2000_0000 to 0x2000_03FF. Store the program 0x2000_0400 through the end address of RAM. The start address of the program must be even address.
Internal flash memory	The following addresses are assigned for storing software ID information and passwords. Storing program in below addresses is not recommendable. 0x3F81_FFF0 to 0x3F81_FFFF

Note: If a password is erased data (0xFF), it is difficult to protect data secure due to an easy-to-guess password. Even if the single boot mode is not used, it is recommended to set a unique value as a password.

23.3.4 Operation Command

The boot program provides the following operation commands.

Table 23-12 Operation command data

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory chip erase and protect bit erase

23.3.4.1 RAM Transfer

The RAM transfer is to store data from the controller to the built-in RAM . When the transfer is complete normally, a user program starts. User program can use the memory address of 0x2000_0400 or later except 0x2000_0000 to 0x2000_03FF for the boot program. CPU will start execution from RAM store start address. The start address must be even address.

This RAM transfer function enables user-specific on-board programming control. In order to execute the on-board programming by a user program, use Flash memory command sequence explained in 23.2.6.

23.3.4.2 Flash Memory Chip Erase and Protect Bit Erase

Flash memory chip erase and protect bit erase commands erase the entire blocks of Flash memory and write/erase protects of all blocks regardless of write/erase protect or security status.

23.3.5 Common Operation regardless of Command

This section describes common operation under the boot program execution.

23.3.5.1 Serial Operation Mode Determination

When the controller communicates via UART, set the 1st byte to 0x86 at the desired baud rate. Figure 23-6shows waveforms in each case.

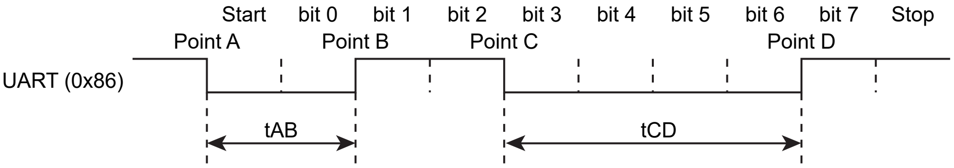


Figure 23-6 Serial operation mode determination data

Figure 23-7 shows a flowchart of boot program. Using 16-bit timer (TMRB) with the time of tAB, tAC and tAD, the 1st byte of serial operation mode determination data (0x86) after reset is provided. In Figure 23-7, the CPU monitors level of the receive pin, and obtains a timer value at the moment when the receive pin's level is changed. Consequently, the timer values of tAB, tAC and tAD have a margin of error. In addition, note that if the transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin.

The flowchart in Figure 23-8shows the serial operation mode is determined that the time length of the receive pin is long or short. If the length is tAB ≤ tCD, the serial operation mode is determined as UART mode. The time of tAD is used whether the automatic baud rate setting is enable or not. Note that timer values of tAB, tAC and tAD have a margin of error. If the baud rate is high and operation frequency is low, each timer value becomes small. This may generates unexpected determination occurs. (To prevent this problem, re-set UART within the programming routine.)

For example, When UART mode is utilized, the controller should allow for a time-out period where the time is expected to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time.

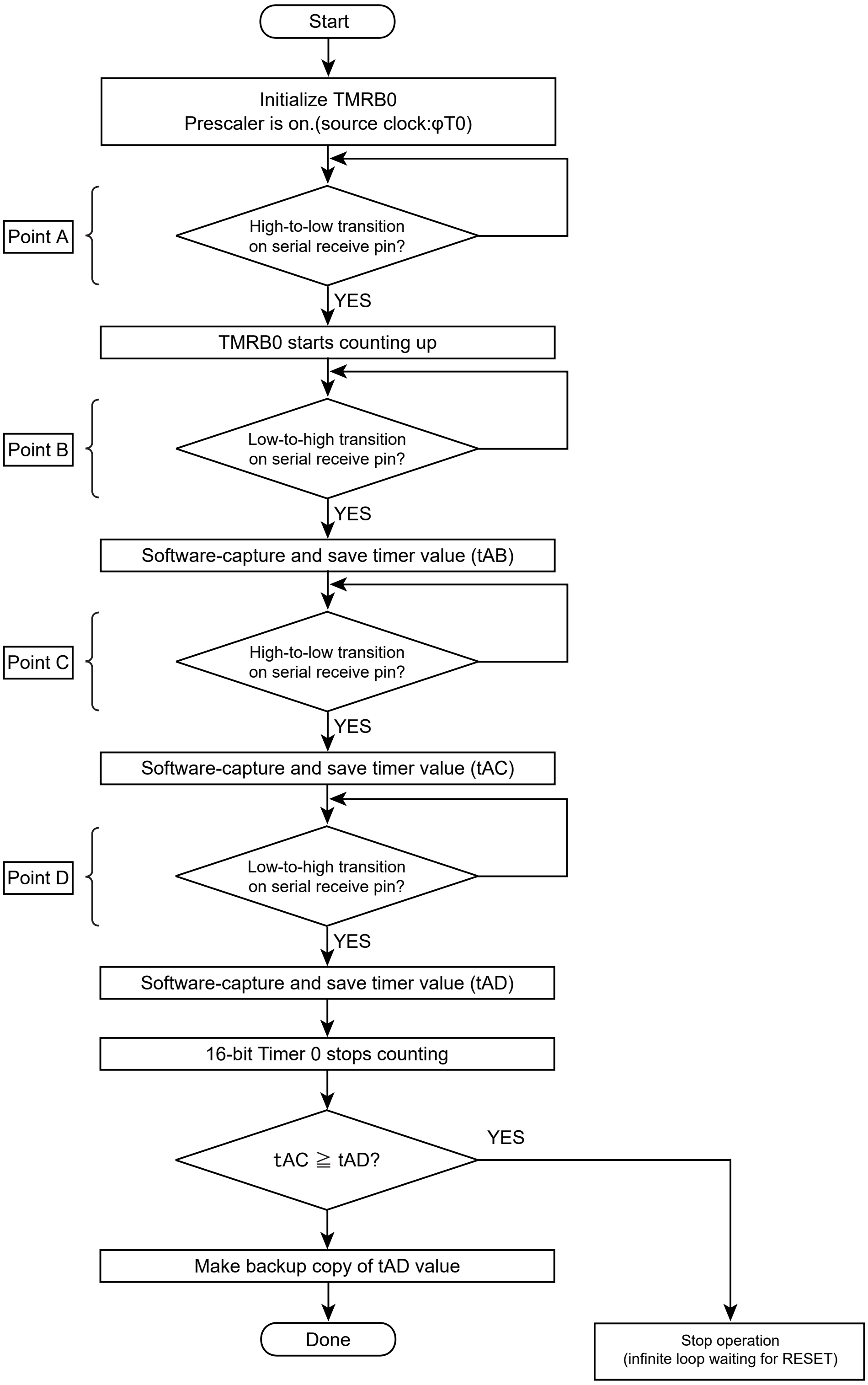

Start
Initialize TMRB0
Prescaler is on.(source clock:φT0)
Point A
High-to-low transition on serial receive pin?
YES
TMRB0 starts counting up
Point B
Low-to-high transition on serial receive pin?
YES
Software-capture and save timer value (tAB)
Point C
High-to-low transition on serial receive pin?
YES
Software-capture and save timer value (tAC)
Point D
Low-to-high transition on serial receive pin?
YES
Software-capture and save timer value (tAD)
16-bit Timer 0 stops counting
tAC ≧ tAD?
YES
Make backup copy of tAD value
Done
Stop operation
(infinite loop waiting for RESET)

Figure 23-7 Serial operation mode receive flowchart

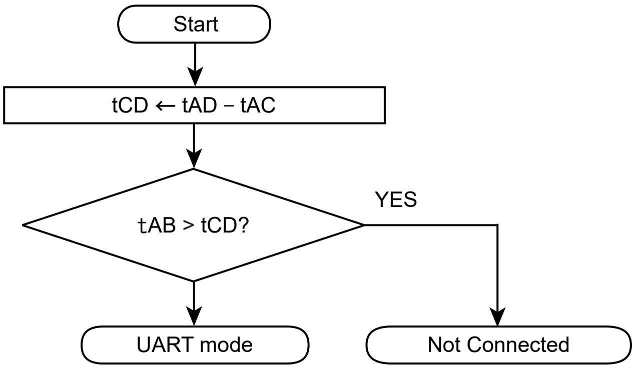


Figure 23-8 Serial operation mode determination flowchart

23.3.5.2 Acknowledge Response Data

The boot program represents processing states in specific codes and sends them to the controller. Table 23-13 to Table 23-16 show the values of acknowledge responses to each receive data.

In Table 23-14 to Table 23-16, the upper four bits of the acknowledge response are equal to those of the operation command data. The 3rd bit indicates a receive error. The 0th bit indicates an invalid operation command error, a checksum error or a password error. The 1st bit and 2nd bit are always "0". Receive error checking is not performed in I/O Interface mode.

Table 23-13 ACK response to the serial operation determination data

Transmit data	Description
0x86	Determined that UART communication is possible. (Note)

Note: When the serial operation is determined as UART, if the baud rate setting is determined as unacceptable, the boot program aborts without sending back any response.

Table 23-14 ACK response to the operation command data

Transmit data	Description
0xN8 (Note)	A receive error occurs in the operation command data
0xN1 (Note)	An undefined operation command data is received normally.
0x10	Determined as a RAM transfer command
0x40	Determined as a flash memory chip erase command

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 23-15 ACK response to the CHECK SUM data

Transmit data	Description
0xN8 (Note)	A receive error occurs.
0xN1 (Note)	A CHECK SUM or a password error occurs.
0xN0 (Note)	The CHECK SUM value is correct.

Note: The upper 4 bits of the ACK response data are the same as those of the operation command data.

Table 23-16 ACK response to Flash memory chip erase and protect bit erase operation

Transmit data	Description
0x54	Determined as a erase enable command
0x4F	Erase command is complete.
0x4C	Erase command is abnormally terminated.

Note:Even when an erase command is performed normally, a Negative acknowledge may be returned by ACK response. Check the FCSR<RDY/BSY> to make sure the command sequence end , and then hold for 200 μ s or more, after that reconfirm the erase status.

23.3.5.3 Password Determination

The boot program use the below area to determine whether a password is required or use as a password.

Area	Address
Password requirement determination	0x3F80_FFF0 (1byte)
Password area	0x3F80_FFF4 to 0x3F80_FFFF (12byte)

The RAM Transfer command performs a password verification regardless of necessity judging data. Flash memory chip erase or protect bit erase command performs a password verification only when necessity judging is determined as "required".

Password requirement setting	Data
Need password	Other than 0xFF
No password	0xFF

If a password is set to 0xFF (erased data), it is difficult to protect data securely due to an easy-to-guess password. Even if Single Boot mode is not used, it is recommended to set a unique value as a password.

(1) Password verification using RAM transfer command

If all these address locations contain the same bytes of data other than 0xFF, this condition is determined as a password area error as shown in Figure 23-9. In this case, the boot program returns an error acknowledge (0x11) in response to the 17th byte of checksum value regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

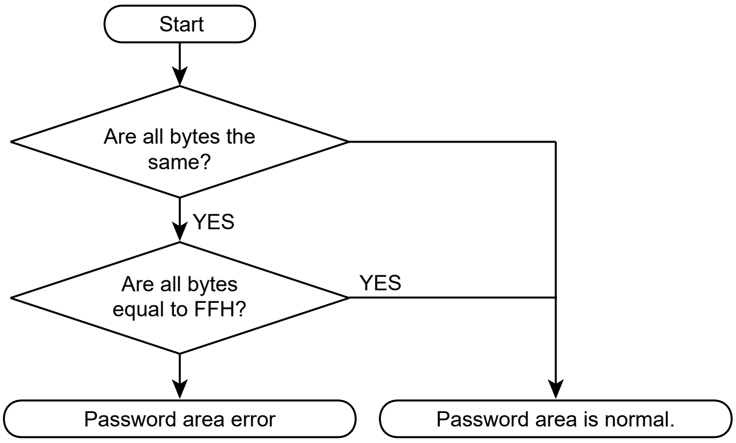


Figure 23-9 Password area check flowchart

(2) Password verification to Flash memory chip erase and protect bit erase command

When a password is enable in the erase password necessity determination area as shown in Figure 23-10 and the passwords are identical data, a password area error occurs. If a password area error is determined, an ACK response to the 17th byte of CHECK SUM sends 0x41 regardless of the password verification.

The boot program verifies 5th byte to 16th byte of receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response data to the 17th of CHECK SUM data is a password error.

The password verification is performed even if the security function is enabled.

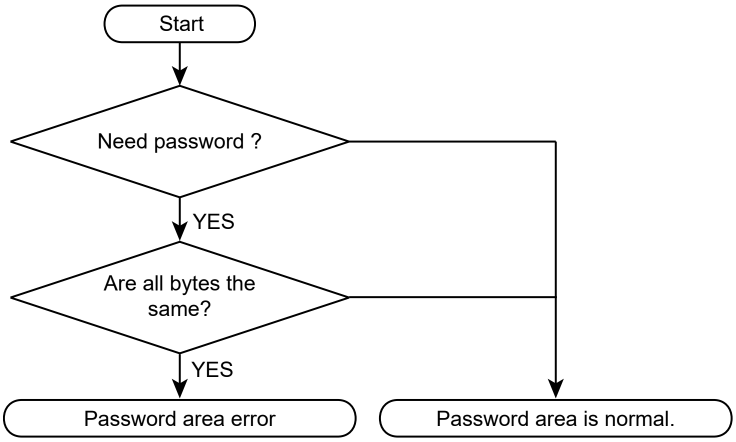


Figure 23-10 Password area check flowchart

23.3.5.4 CHECK SUM Calculation

The checksum is calculated by 8-bit addition to transmit data, dropping the carries, and taking the two’s complement of the total sum. The controller must perform the same checksum operation in transmitting checksum bytes.

Example of CHECK SUM

To calculate the checksum for a series of 0xE5 and 0xF6, perform 8-bit addition.

0xE5 + 0xF6 = 0x1DB

Take the two’s complement of the sum to the lower 8-bit, and that is a checksum value. So the boot program sends 0x25 to the controller.

0 − 0xDB = 0x25

23.3.6 Transfer Format at RAM Transfer

This section shows a RAM transfer command format. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM37AFSQG

Transfer direction (C←T): TMPM37AFSQG to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to "23.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, the program determines whether a baud setting is possible. If not, the program stops and communication is shutdown.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
3	C→T	Operation command data (0x10)	Sends RAM transfer command data (0x10).
4	C←T	ACK response to operation command Normal state: 0x10 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 23-12, receive data is echoed back. In the case of RAM transfer, 0x10 is echoed back and the transfer data branches to the RAM transfer service routine. If the data does not correspond to the command in Table 23-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.)
5 to 16	C→T	Password data (12-byte) 0x3F80_FF04 to 0x3F80_FF0F	Checks data in the password area. For detail of password area checking, refer to "23.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F81_FFF0 to 0x3F81_FFFF of data of Flash memory. If the data does not match the address, a password error flag is set.
17	C→T	5th to 16th byte of CHECK SUM values	Send 5th to 16th byte of CHECK SUM values. For detail of CHECK SUM calculation, refer to 23.3.5.4.

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 5th to 17th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 17th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x11 that means a password error and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
19	C→T	RAM store start Address 31 to 24	Sends a start address of block transfer for RAM store. The 19th byte corresponds to 31st to 24th bit of address. The 22nd byte corresponds to 7th to 0th bit of address. Specify the address to the address 0x2000_0400 through the last address of RAM. The address must be even address.
20	C→T	RAM store start Address 23 to 16	
21	C→T	RAM store start Address 15 to 8	
22	C→T	RAM store start Address 7 to 0	
23	C→T	Number of RAM store bytes 15 to 8	Set the number of bytes to perform block transfer. The 23rd byte corresponds to the15th bit to 8th bit of transfer bytes. The 24th byte corresponds to 7th bit to 0th bit of transfer bytes. Specify the data to be stored in the address from 0x2000_0400 through the last address of RAM.
24	C→T	Number of RAM store bytes 7 to 0	
25	C→T	19th to 24th byte of CHECK SUM value	Send 19th byte to 24th byte of CHECK SUM values
26	C←T	ACK response to CHECK SUM value Normal state: 0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 19th byte to 25th byte of receive data have errors.(UART mode only) If receive errors exist, sends a ACK response data 0x18 that means abnormal communications and waits for a next operation command (3rd byte). Then checks 25th byte of CHECK SUM data. If errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
27 to m	C→T	RAM stored data	Sends same bytes of data specified in 23th bytes to 24 byte for RAM stored data.
m+1	C→T	27 to m byte of CHECK SUM value	Sends 27th byte to m byte of CHECK SUM value
m+2	C←T	ACK response to CHECK SUM value Normal state:0x10 Abnormal state: 0x11 Communication error: 0x18	First, checks if 27th byte to m+1 byte of receive data have errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Then checks m+1 byte of CHECK SUM data, if errors exist, sends 0x11 and waits for a next operation command (3rd byte). If all procedure normally ends, sends a normal ACK response data 0x10.
–	–	–	If m + 2nd byte of ACK response data is normal ACK response data, the transfer data branches to the address specified in 19th byte to 22 byte.

23.3.7 Transfer Format of Flash memory Chip Erase and Protect Bit Erase

This section shows a transfer format of Flash memory chip erase and protect bit erase commands. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM37AFSQG

Transfer direction (C←T): TMPM37AFSQG to Controller

Number of transfer bytes	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Sends data to determine the serial operation mode. For detail of mode determination, refer to"23.3.5.1 Serial Operation Mode Determination".
		[UART mode] 0x86	Sends 0x86. If UART mode is determined, checks if baud rate setting can be done. If not, operation stops communications.
2	C←T	ACK response to serial operation mode	The 2nd byte of transmit data is a ACK response data to the 1st byte that corresponds to the serial operation setting mode data. If the setting is possible, sets SIO/UART. A receive enable timing is set before transmit buffer is written to the data.
		[UART mode] Normal state: 0x86	If the setting is determined to be possible, sends 0x86. If not, the operation aborts without sending back any response. When the controller finished to send the 1st byte of data, requires a time-out time (5 seconds). If data (0x86) is not normally received within a time-out time, communication is not possible.
3	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
4	C←T	ACK response to the operation command Normal state: 0x40 Abnormal state: 0xX1 Communication error: 0xX8	ACK response data to the operation command. First, checks if 3rd byte of receive data has errors. (UART mode only) If receive errors exist, sends a ACK response data 0xX8 that means abnormal communications and waits for a next operation command (3rd byte). Upper 4 bits of transmit data are undefined. (same as upper 4 bits of immediately before operation command data.) Note that in the I/O interface, receive error check is not performed. Then, if the 3rd byte of receive data corresponds to either operation command data in Table 23-12, receive data is echoed back. If the data does not correspond to the command in Table 23-12, sends a ACK response data 0xX1 that means operation command errors, and waits for next operation command. (3rd byte) Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
5 to 16	C→T	Password data (12-byte) 0x3F80_FF04 to 0x3F80_FF0F	If password necessity is set to "none", this data is dummy data. If password necessity is set to "necessary", checks data in the password area. For a method of password area data checking, refer to "23.3.5.3 Password Determination". Compares 5th to 16th byte of receive data with 0x3F80_FFF0 to 0x3F80_FFFF of data of Flash memory in order. If the data does not match, a password error flag is set.
17	C→T	5 to 16 th byte CHECK SUMvalue	Sends 5th byte to 16 byte of CHECK SUM value. For a method of CHECK SUM calculation, refer to"23.3.5.4 CHECK SUM Calculation".

Number of transfer bytes	Transfer direction	Transfer data	Description
18	C←T	ACK response to the CHECK SUM value Normal state: 0x40 Abnormal state: 0x41 Communication error: 0x48	If password necessity is set to "none", sends a normal ACK response data 0x40. If password necessity is set to "necessary", first checks if receive errors exist in the 5th byte to 17th byte receive data. (UART mode only) If a receive error exists, sends a ACK response data 0x48 that means abnormal communications and waits for next operation command. (3rd byte) Then checks 17th byte of CHECK SUM data. If error occurs, sends 0x41 and waits for a next operation command (3rd byte) Finally, checks the result of password verification. If a password error exists, sends a ACK response data 0x41that means a password error and waits for a next operation command (3rd byte) If all procedure normally ends, sends a normal ACK response data 0x40.
19	C→T	Erase enable command data (0x54)	Sends an enable command data (0x54).
20	C←T	ACK response to the erase enable command Normal state: 0x54 Abnormal state: 0xX1 Communication error: 0x58	First, checks if 19th byte of receive data has errors. If receive errors exist, sends a ACK response data (bit 3) 0x58 that means abnormal communication and waits for next operation command (3rd byte). Then, if 19th byte of receive data corresponds to the erase enable command, receive data is echoed back (normal ACK response data). In this case, 0x54 is echoed back and the transfer data branches into Flash memory chip erase process routine. If the data does not correspond to the erase enable command, sends a ACK response data (bit 0) 0xX1 and waits for next operation command. Upper 4 bits of transmit data are undefined. (Upper 4 bits of immediate before operation command data are used.)
21	C→T	ACK response to the erase command (note1) Normal state: 0x4F Abnormal state: 0x4C	If the operation is normally complete, the end code (0x4F) is returned. If erase error occurs, an error code (0x4C) is returned.
–	–	–	Waits for a next operation command.

Note 1: Even when an erase command is performed normally, a Negative acknowledge may be returned by ACK response. Check the FCSR<RDY/BSY> to make sure the command sequence end , and then hold for 200 μ s or more, after that reconfirm the erase status.

23.3.8 Boot Program Whole Flowchart

This section shows a boot program whole flowchart.

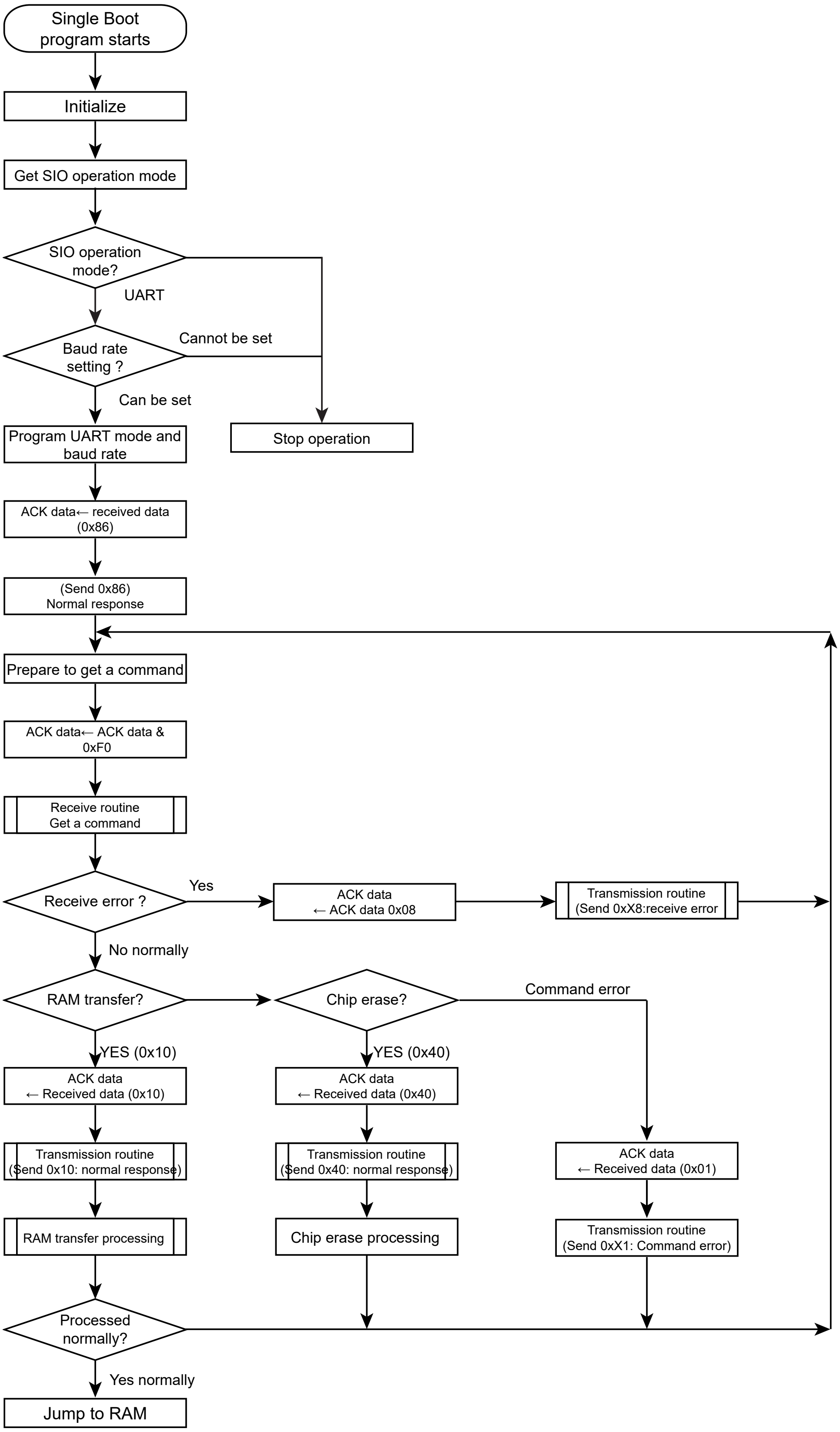

Single Boot program starts
Initialize
Get SIO operation mode
SIO operation mode?
UART
Baud rate setting ?
Cannot be set
Can be set
Program UART mode and baud rate
Stop operation
ACK data← received data (0x86)
(Send 0x86) Normal response
Prepare to get a command
ACK data← ACK data & 0xF0
Receive routine Get a command
Receive error ?
Yes
ACK data ← ACK data 0x08
Transmission routine (Send 0xX8:receive error
No normally
RAM transfer?
Chip erase?
Command error
YES (0x10)
YES (0x40)
ACK data ← Received data (0x10)
ACK data ← Received data (0x40)
Transmission routine (Send 0x10: normal response)
Transmission routine (Send 0x40: normal response)
ACK data ← Received data (0x01)
RAM transfer processing
Chip erase processing
Transmission routine (Send 0xX1: Command error)
Processed normally?
Yes normally
Jump to RAM

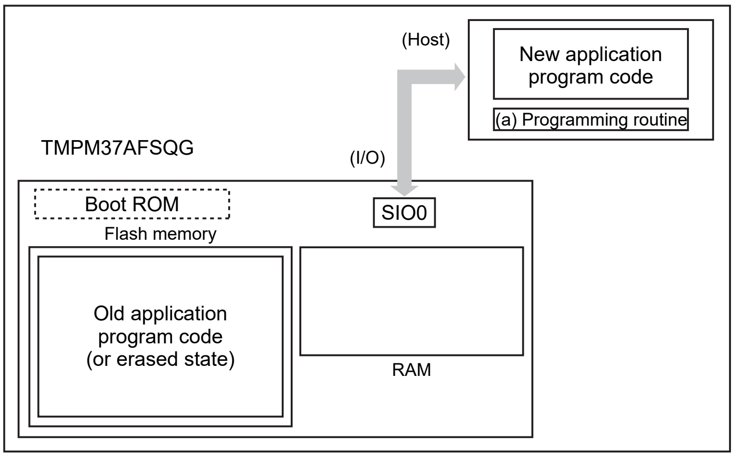
Figure 23-11 Boot program whole flowchart

23.3.9 Reprogramming Procedure of Flash using reprogramming algorithm in the on-chip BOOT ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip boot ROM.

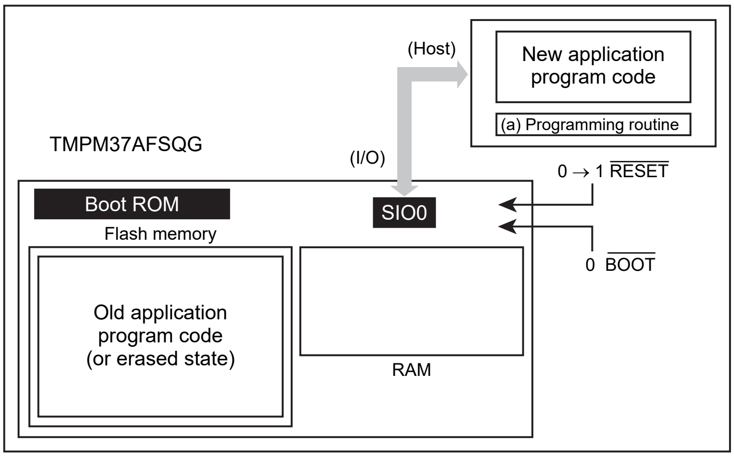
23.3.9.1 Step-1

The condition of Flash memory does not care whether a user program made of former versions has been written or erased. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to an external host. A programming routine (a) is prepared on the host.


(Host)
New application program code
(a) Programming routine
TMPM37AFSQG
(I/O)
Boot ROM
Flash memory
SIO0
Old application program code (or erased state)
RAM

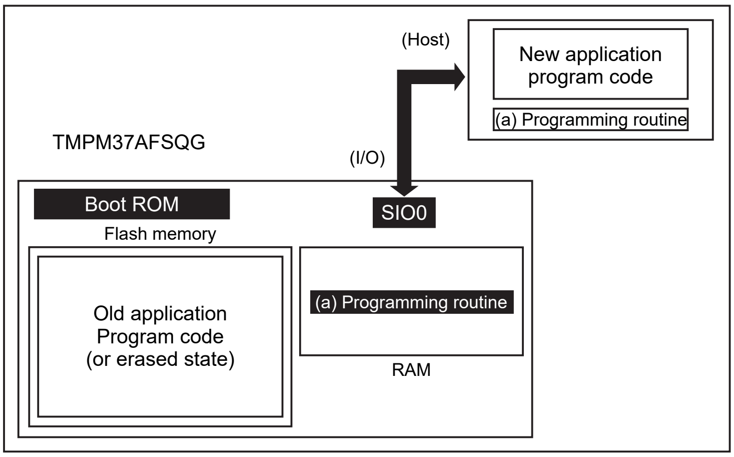
23.3.9.2 Step-2

Release the reset by pin condition setting in the boot mode and boot-up the BOOT ROM. According to the procedure of boot mode, transfer the programming routine (a) via SIO0 from the source (host). A password verification with the password in the user application program is performed. (If Flash memory is erased, an erase data (0xFF) is dealt with a password.)


(Host)
New application program code
(a) Programming routine
TMPM37AFSQG
(I/O)
0 → 1 $\overline{\text{RESET}}$
Boot ROM
Flash memory
SIO0
0 $\overline{\text{BOOT}}$
Old application program code (or erased state)
RAM

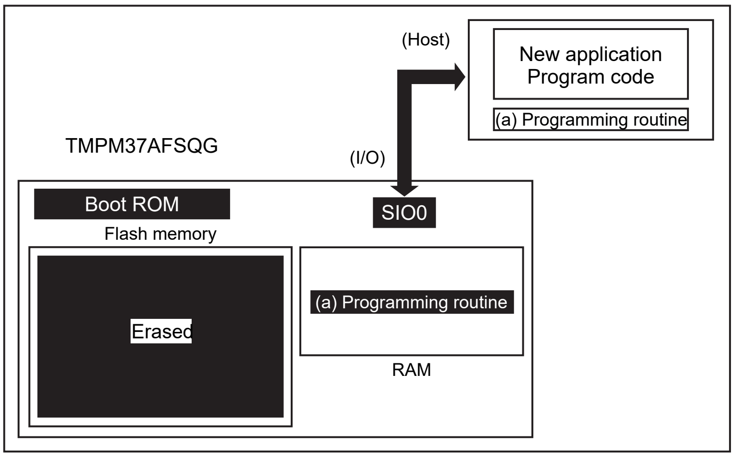
23.3.9.3 Step-3

If the password verification is complete, the boot program transfer a programming routine (a) from the host into the on-chip RAM. The programming routine must be stored in the range from 0x2000_0400 to the end address of RAM.



23.3.9.4 Step-4

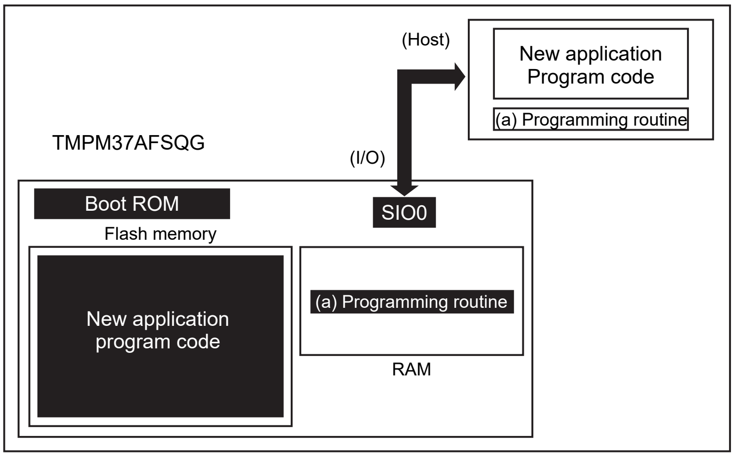
The boot program jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing old application program codes. The Block Erase or Chip Erase command is be used.



23.3.9.5 Step-5

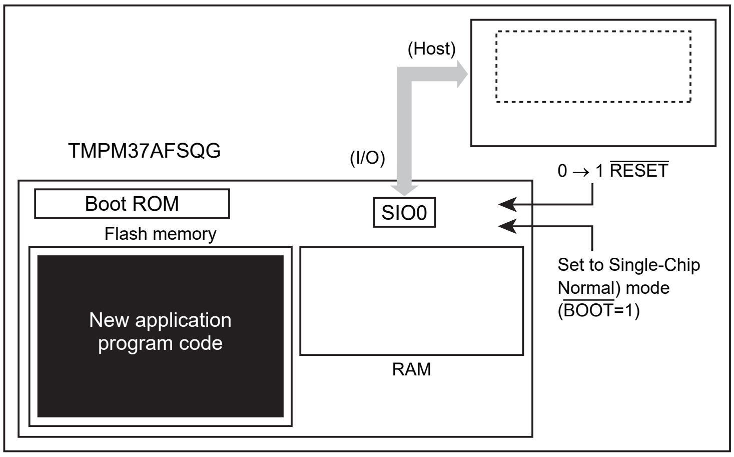
The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is complete, the writing or erase protection of that flash area in the user's program must be set.

In the example below, new program code comes from the same host via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create a hardware board and programming routine to suit your particular needs.



23.3.9.6 Step-6

When programming of Flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the device re-boots in the single-chip (Normal) mode to execute the new program.



23.4 Programming in the User Boot Mode

A user Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the user application is different from the serial I/O. It operates in the single chip mode; therefore, a switch from normal mode in which user application is activated in the use boot mode to the user boot mode for programming flash is required. Specifically, add a mode judgment routine to the reset service routine in the user application program.

The condition to switch the modes needs to be set according to the user's system setup condition. Also, a flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to the user boot mode. The data in built-in Flash memory cannot be read out during erase/reprogramming mode. Thus, reprogramming routine must be take place while it is stored in the area outside of Flash memory area. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental reprogramming. Be sure not to generate interrupt/fault except reset to avoid abnormal termination during the user boot mode.

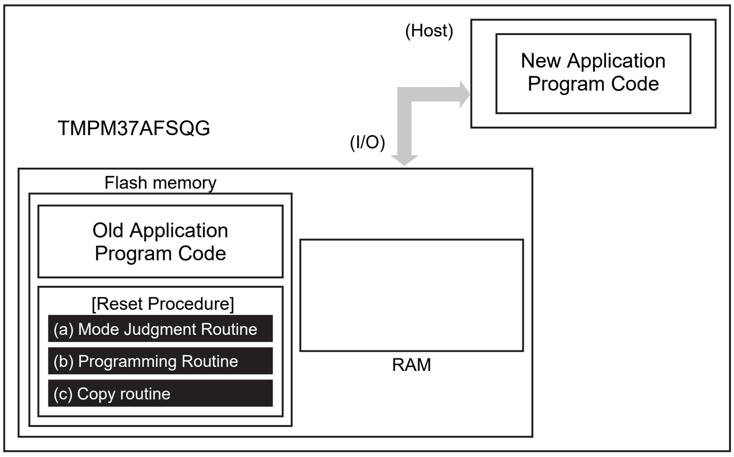
Taking examples from two cases such as the method that reprogramming routine stored in Flash memory (1-A) and transferred from the external device (1-B), the following section explains the procedure. For a detail of the program/erase to Flash memory, refer to "23.2 Detail of Flash Memory".

23.4.1 (1-A) Procedure that a Programming Routine Stored in Flash memory

23.4.1.1 Step-1

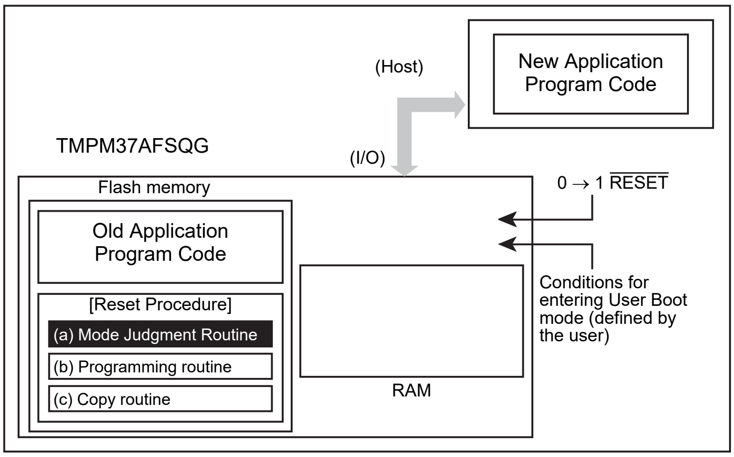
A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following three program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to user boot mode or not
(b) Flash programming routine:	A program to download new program from the host controller and re-program Flash memory
(c) Copy routine:	A program to copy the data described in (a) to the built-in RAM or external memory device



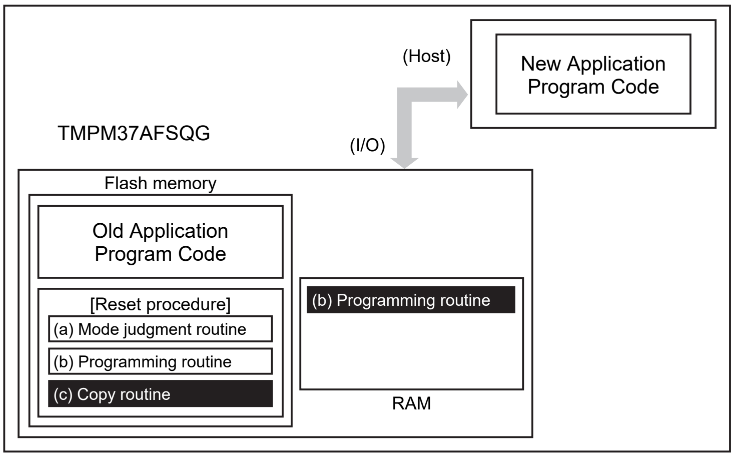
23.4.1.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.


(Host)
New Application Program Code
TMPM37AFSQG
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old Application Program Code
[Reset Procedure]
(a) Mode Judgment Routine
(b) Programming routine
(c) Copy routine
RAM
Conditions for entering User Boot mode (defined by the user)

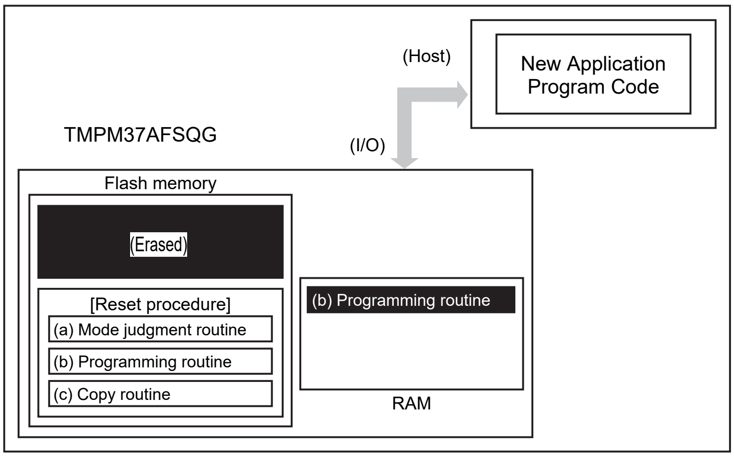
23.4.1.3 Step-3

Once the device enters the user boot mode, execute the copy routine (C) to download the flash programming routine (b) from the host controller to the built-in RAM .



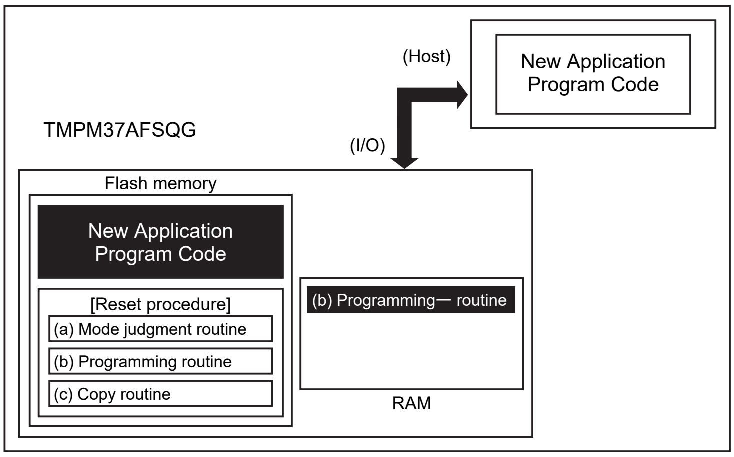
23.4.1.4 Step-4

Jump to the the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



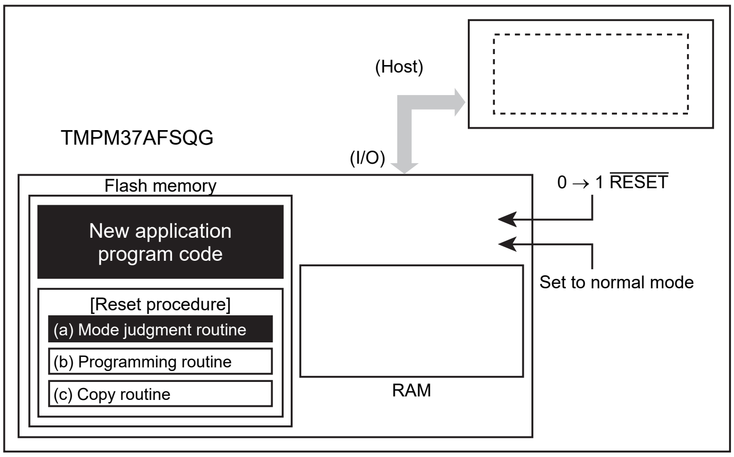
23.4.1.5 Step-5

Continue to execute the flash programming routine to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



23.4.1.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



23.4.2 (1-B) Procedure that a Programming Routine is transferred from External Host

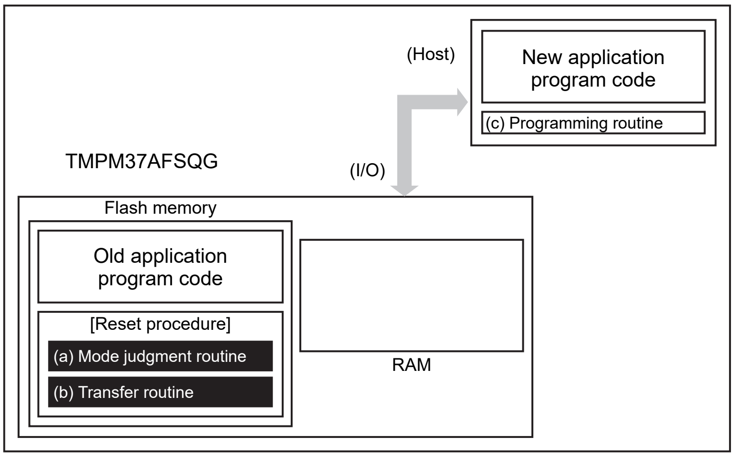
23.4.2.1 Step-1

A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following two program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to reprogramming operation
(b) Transfer routine:	A program to obtain a reprogramming program from the external device.

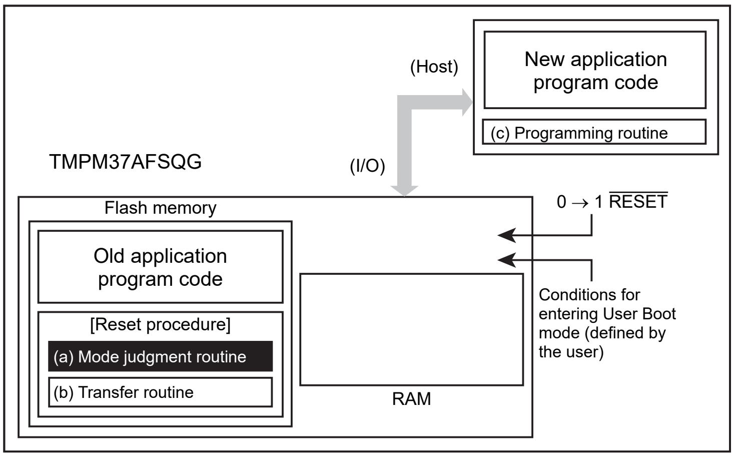
In addition, prepare a reprogramming routine shown below must be stored on the host controller.

(c) Reprogramming routine:	A program to reprogram data



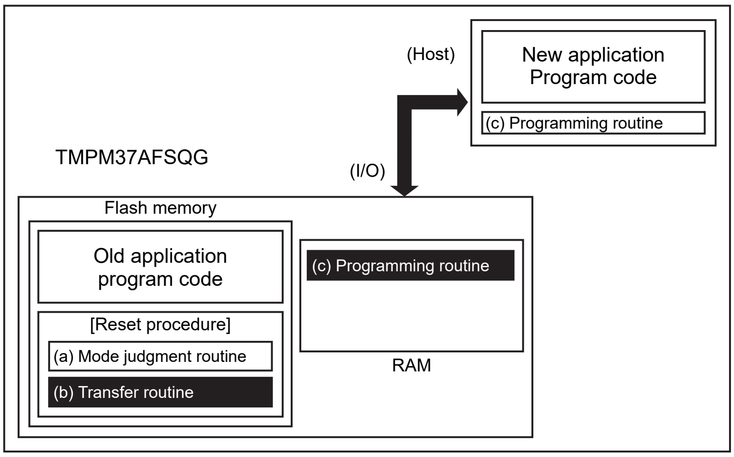
23.4.2.2 Step-2

This section explains the case that a programming routine storied in the reset routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data.


(Host)
New application program code
(c) Programming routine
TMPM37AFSQG
(I/O)
Flash memory
0 → 1 $\overline{\text{RESET}}$
Old application program code
[Reset procedure]
(a) Mode judgment routine
(b) Transfer routine
RAM
Conditions for entering User Boot mode (defined by the user)

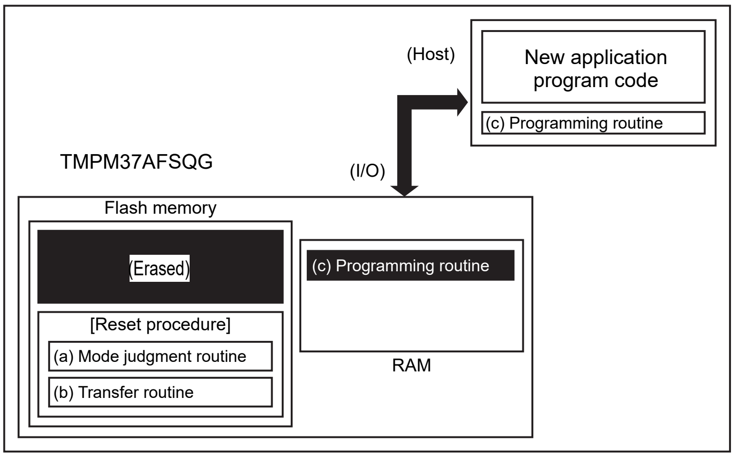
23.4.2.3 Step-3

Once the device enters the user boot mode, execute the transfer routine (b) to download the programming routine (c) from the host controller to the built-in RAM .



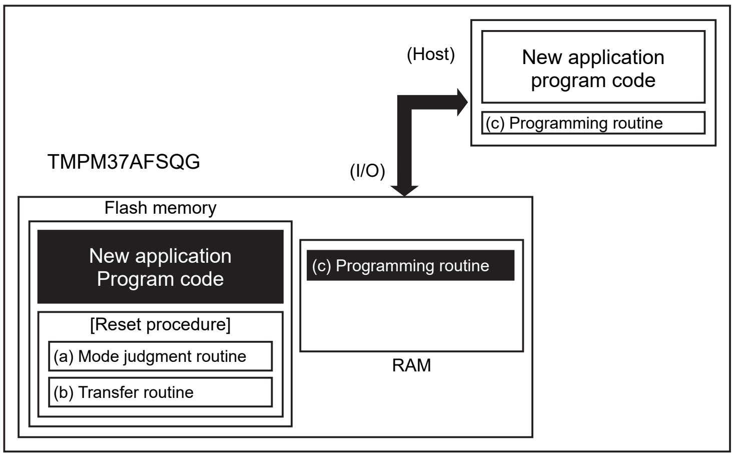
23.4.2.4 Step-4

Jump to the the reprogramming routine in the built-in RAM to release the write/erase protection for the old application program, and to erase a flash in block unit.



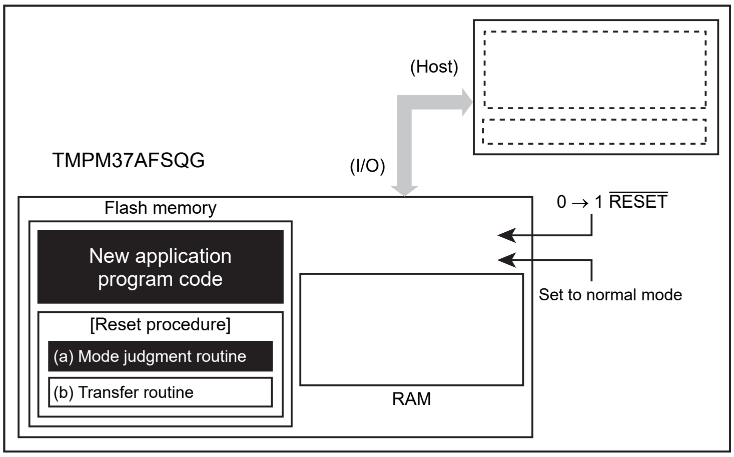
23.4.2.5 Step-5

Continue to execute the flash programming routine (c) to download new program data from the host controller and program it into the erased flash block. When the programming is complete, the write/erase protection of that flash block in the user program area must be set.



23.4.2.6 Step-6

Set $\overline{\text{RESET}}$ to "0". Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



24. Debug Interface

24.1 Specification Overview

The TMPM37AFSQG contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the Debug interface. For more information of SWJ-DP, ETM and TPIY, please refer to the "ARM documentations set for the Cortex-M3" issued by ARM Limited.

24.2 Features of SWJ-DP

SWJ-DP supports the two-pin Serial Wire Debug Port (SWCLK, SWDIO).

24.3 Pin Functions

The debug interface pins can also be used as general-purpose ports.

Table 24-1 SWJ-DP, ETM function

SWJ-DP	Name of	SW debug	
Pin name	port	I/O	Description
SWDIO	PB3	I/O	Serial Wire Data Input/Output
SWCLK	PB4	Input	Serial Wire Clock

After reset, the PB3 and PB4 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 24-2 below summarizes the debug interface pin functions and related port settings after reset.

Table 24-2 Debug interface pins and port setting after reset

Initial	Port	Debug	Port Setting After Reset (-;No register)					
Setting	(Bit name)	Function	Function (PBFR)	Input (PBIE)	Output (PBCR)	Open Drain (PBOD)	Pull-up (PBPUP)	Pull-down (PBPDN)
DEBUG	PB3	TMS/SWDIO	1	1	1	0	1	0
DEBUG	PB4	TCK/SWCLK	1	1	0	0	0	1

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

Note 1: If PB3 is 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.

24.4 Connection with a Debug Tool

24.4.1 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.

24.4.2 Notes When Using the General-purpose Ports

When debugging, do not change setting of the debugging interface connecting to general-purpose ports by program; otherwise the MCU cannot receive the control signals from the debugging tools and continue to debugging. According to the usage of the debugging interface pins, check their settings.

Table 24-3 Debug Interface

Usage	Using Debug Interface	
	SWCLK	SWDIO
SW (After RESET)	O	O
SW	O	O

O:Enable

-:Disable (Using the General-purpose Ports)

24.5 Peripherals operation during HALT mode

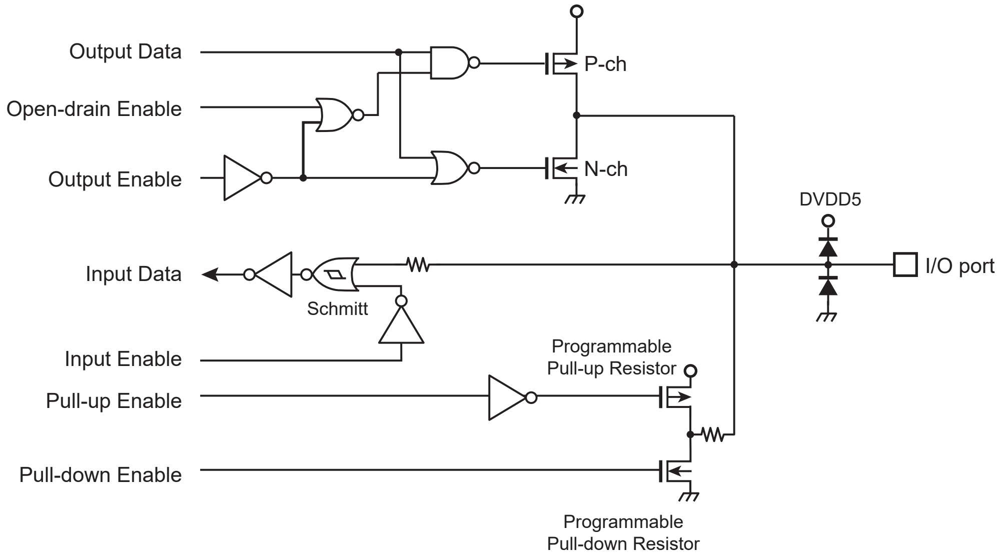
When Break during debugging, Cortex-M3 CPU core going into HALT mode. Watch dog timer (WDT) is stopped counting automatically. And 16bit timer/counter can specify the status (continue operating or stop) in HALT mode. Other peripherals are continue operating.

25. Port Section Equivalent Circuit Schematic

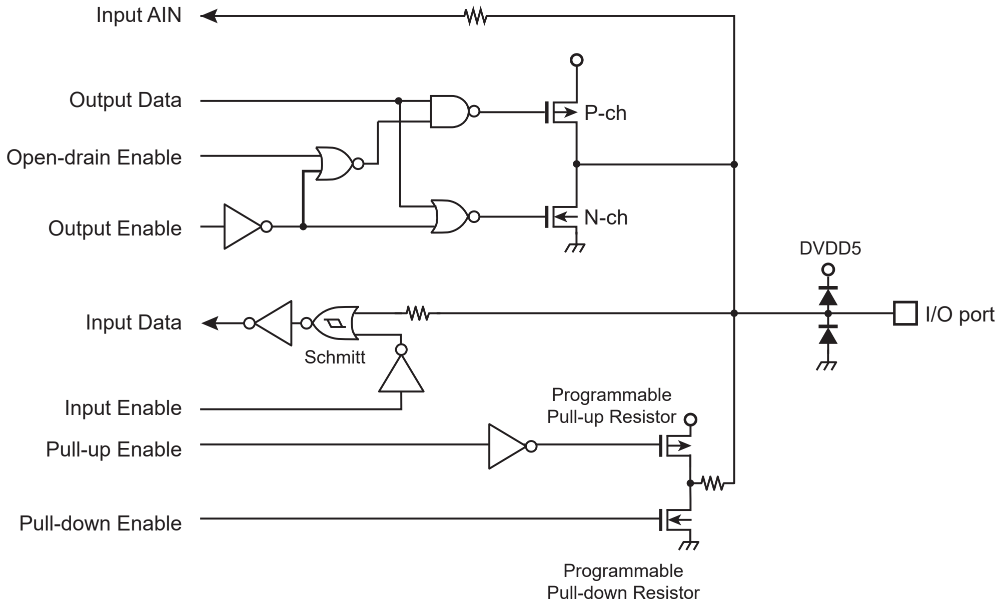
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of Ω to several hundreds of Ω. Feedback resistor and Damping resistor are shown with a typical value.

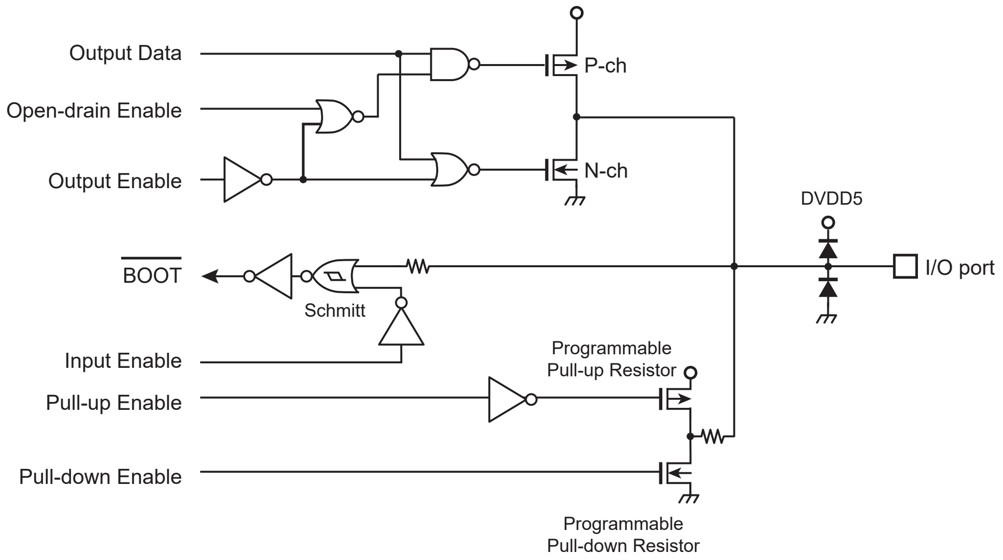
25.1 PB3 to 4, PE0 to 2



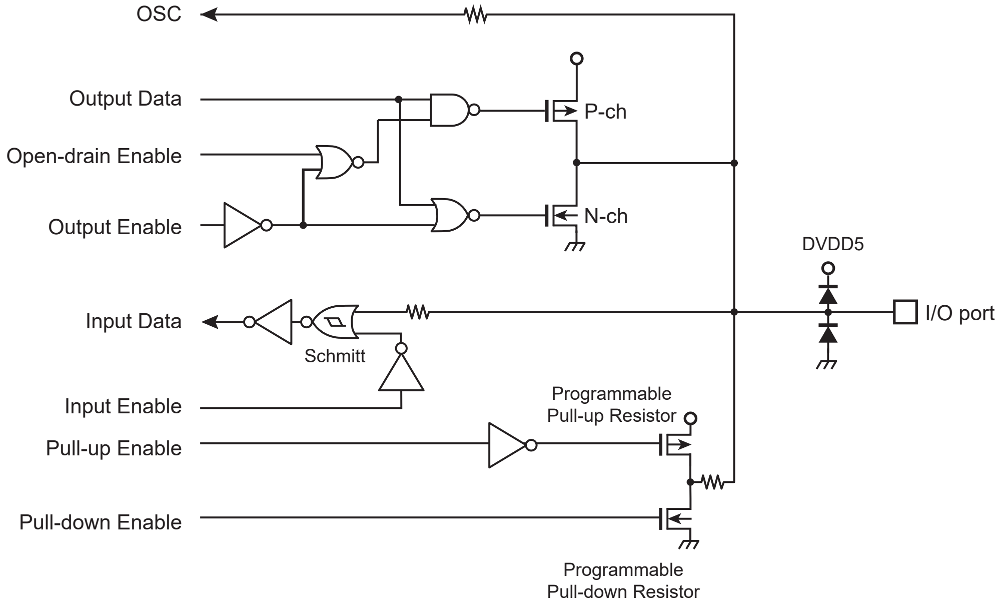
25.2 PJ5 to 7, PK0 to 1



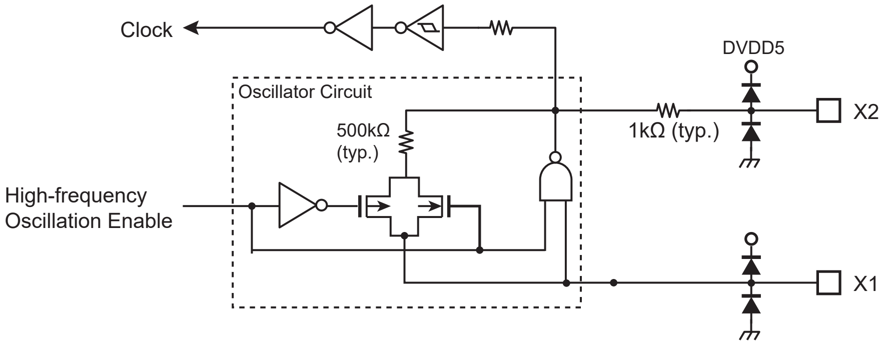
25.3 PF0



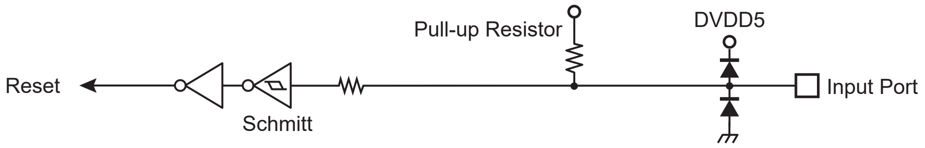
25.4 PM0 to 1



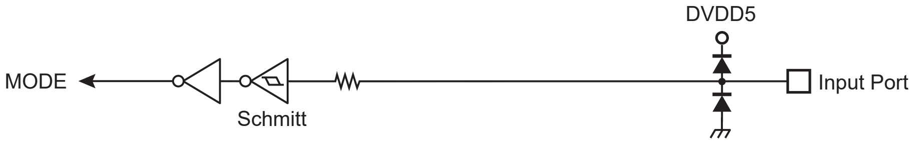
25.5 X1, X2



25.6 $\overline{\text{RESET}}$

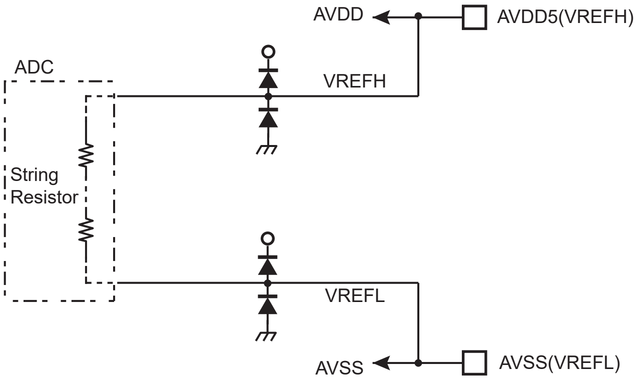


25.7 MODE

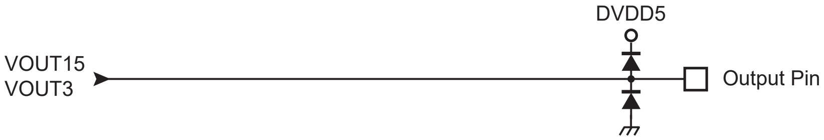


(Note)MODE pin is fixed to GND.

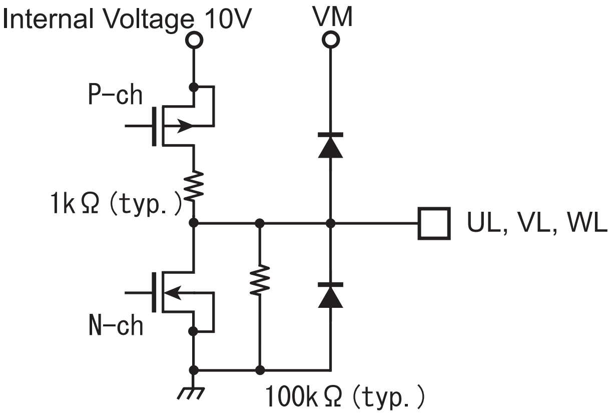
25.8 VREFHB, VREFLB



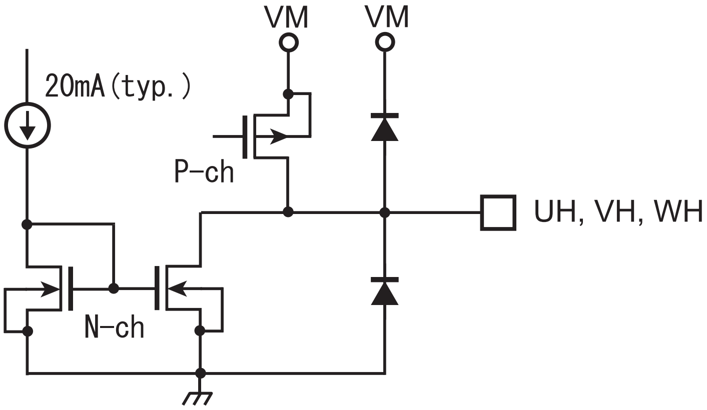
25.9 VOUT15, VOUT3



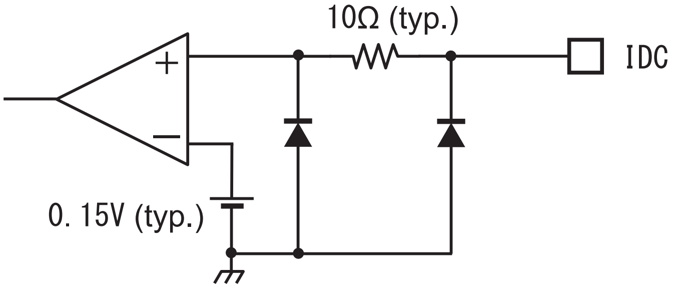
25.10 UL, VL, WL



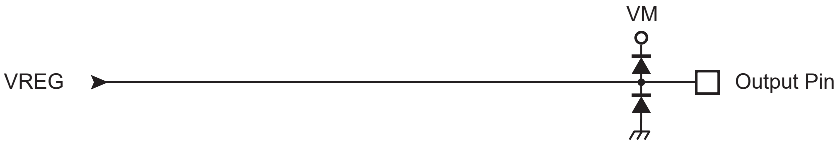
25.11 UH, VH, WH



25.12 IDC



25.13 VREG



26. Electrical Characteristics

26.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5B / AVDD5B / AMPVDD5	−0.3 to 6	V
		VM	40	
Capacitor voltage		VOUT15	−0.3 to 3	V
		VOUT3	−0.3 to 3.9	
Input voltage		V_{IN}	−0.3 to VDD + 0.3	V
Output voltage	UH,VH,WH	V_{OUT}	40	V
	UL,VL,WL		15	
Output current	UH,VH,WH	I_{OUT}	30	mA
	UL,VL,WL		-30	
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	-50	
Power consumption		PD	350	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−55 to 125	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	−40 to 85	°C
	During Flash W/E		0 to 70	

Note: **Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.**

Note: **VDD = DVDD5B / AVDD5B / AMPVDD5**

26.2 DC Electrical Characteristics (1/2)

DVSS = DVSSB = AVSSB = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Rating		Min	Typ. (Note 1)	Max	Unit
Supply voltage (Note 2)	DVDD5B AVDD5B AMPVDD5	VDD	f_{OSC} = 8 or 10 MHz fsys = 1 to 40 MHz		4.5	–	5.5	V
Supply voltage (during Flash W/E) (Note 2)	DVDD5B AVDD5B AMPVDD5	VDD	f_{OSC} = 8 or 10 MHz fsys = 1 to 40 MHz (Ta (°C) = 0 to 70)		4.5	–	5.5	V
Supply voltage (Power-on or Power-off) (Note 5)	DVDD5B AVDD5B AMPVDD5	VDD	f_{OSC} = 8 or 10 MHz fsys = 1 to 40 MHz		3.9	–	5.5	V
Low-level input voltage	Schmitt-Input	V_{IL1}	VDD = 4.5V to 5.5V (Note 4)		−0.3	–	0.25 VDD	V
High-level input voltage	Schmitt-Input	V_{IH1}	VDD = 4.5V to 5.5V (Note 4)		0.75VDD		VDD+0.3	V
Capacitance for VOUT15 and VOUT3 (Note 3)		C_{out}	VOUT15, VOUT3		3.3	–	4.7	μF
Low-level output voltage		V_{OL}	I_{OL} = 1.6 mA	VDD ≥ 4.5V (Note 4)	–	–	0.4	V
High-level output voltage		V_{OH}	I_{OH} = −1.6 mA	VDD ≥ 4.5V (Note 4)	4.1	–	–	V
Input leakage current		I_{LI1}	0.0 ≤ V_{IN} ≤ VDD (Note 4)		–	0.02	±5	μA
Output leakage current		I_{LO}	0.2 ≤ V_{IN} ≤ VDD -0.2 (Note 4)		–	0.05	±10	
Pull-up resister at Reset		R_{RST}	4.5 ≤ VDD ≤ 5.5 (Note 4)		–	50	150	kΩ
Programmable pull-up/pull-down resistor		P_{KH}	4.5 ≤ VDD ≤ 5.5 (Note 4)		–	50	150	kΩ
Schmitt-Triggered port		V_{TH}	4.5 ≤ VDD ≤ 5.5 (Note 4)		0.3	0.6	–	V
Pin capacitance (Except power supply pins)	Digital pins	C_{IO1}	fc = 1 MHz		–	–	10	pF
	Analog pins PJ6, PJ7, PK0 and PK1	C_{IO2}			–	–	30	

Note 1: Ta = 25 °C, DVDD5B = AVDD5B = 5V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD5B and AVDD5B.

Note 3: VOUT15 and VOUT3 pin should be connected to GND via same value of capacitance. The IC outside cannot have the power supply from VOUT15 and VOUT3.

Note 4: VDD = DVDD5B = AVDD5B

Note 5: It is a voltage range in the case of Power-on or Power-off (when VLTD disabled). In the range whose Power-line is 3.9V ≤ VDD < 4.5V, does not guarantee a 12-bit AD converter and AC electrical Characteristics.

26.3 DC Electrical Characteristics (2/2)

DVDD5B = 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 = 40 MHz	−	20	35	mA
IDLE (Note 3) Gear 1/1			−	4	11	
STOP		−	−	0.2	2	

Note 1: Ta = 25 °C, DVDD5B = AVDD5B = 5V, unless otherwise noted.

Note 2: I_{DD} NORMAL: All functions operates excluding ADC and Op-amp.

Note 3: I_{DD} IDLE : All peripheral functions stopped.

26.4 12-bit ADC Electrical Characteristics

DVDD5B = AVDD5B / VREFHB = 4.5 V to 5.5 V
DVSSB = AVSSB / VREFLB = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Rating	Min	Typ.	Max	Unit
Analog reference voltage (+)		VREFHB	−	−	AVDD	−	V
Analog input voltage		VAIN	−	AVSS	−	AVDD	V
Supply current	AD conversion	−	Include IREF	−	8.0	11.0	mA
INL error		−	AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 μF Conversion time ≥ 2 μs	−	−	-6 to +10	LSB
DNL error				−	−	±5	
Offset error				−	−	±5	
Full-scale error				−	−	±6	
Total error				−	−	-7 to +11	

Note: 1LSB = (AVDD − AVSS)/4096 [V]

Note: AVDD = AVDD5B, AVSS = AVSSB

Note: The characteristic is measured under the condition in which the only ADC is operating.

26.5 Op-Amps Electrical Characteristics

DVDD5B = AVDD5B / VREFHB = 4.5 V ~ 5.5 V
DVSSB = AVSSB / VREFLB = 0V, Ta = −40 ~ 85 °C

Parameter		Symbol	Rating	Min	Typ.	Max	Unit
Gain (Note 1)		VGAIN	−	1.5	−	10	
Input voltage range		VAMPIN	−	(AVDD×0.1)/ VGAIN	−	(AVDD×0.9)/ VGAIN	V
Total amplifier output error		VAMP	−	−6 × VGAIN	−	+6 × VGAIN	mV
Slew rate		Vthr (Note 2)	5pF, VGAIN = ×2.5	2	−	−	V /μs
		Vthf		1	−	−	
Supply current	Op-Amp	−	−	−	4	6	mA

Note:AVDD = AVDD5B = 4.5 to 5.5V, AVSS = AVSSB = 0V

Note 1: Gain can be selected among ×2.5, ×3, ×3.5, ×4, ×6 and ×8 by register setting.

Note 2: Slew rate means a slant till the output of amplifier reaches AVDD-0.001×AVDD.

26.6 Pre-Driver Electrical Characteristics

VM = 12 V
DVDD5B = AVDD5B / VREFHB = 4.5 V ~ 5.5 V
DVSSB = AVSSB / VREFLB = 0V, Ta = −40 ~ 85 °C

Parameter	Symbol	Rating	Min	Typ.	Max	Unit
Power supply voltage of VM	V_{M12}	fsys = 1 to 40 MHz, f_{OSC} = 8 or 10 MHz Power supply from VREG to VINREG5, DVDD5 and AVDD5/VREFH	6	12	15	V
	V_{M24}	fsys = 1 to 40 MHz, f_{OSC} = 8 or 10 MHz	6	12,24	32	
Power supply current of VM	I_{VM12}	V_M =12V, V_{REG} =open, Output Open	−	2.8	4	mA
	I_{VM24}	V_M =24V, V_{REG} =open, Output Open	−	2.8	4	
VREG voltage	V_{REG}	I_{VREG} = -50mA, VM=6 to 15V	4.5	5	5.5	V
VREG rise time	Tr VREG	C_{VREG} =4.7µF, I_{VREG} =50mA	10	−	−	µs
Upper Output current	$I_{OUTP(L)}$	V_{OUTP} =12V	18.4	23	27.6	mA
Upper Output leakage current	$I_{OUTP(H)}$	V_{OUTP} =32V,VM=32V	-1	−	1	µA
Lower Output High voltage	$V_{OUTN(H)}$	I_{OUTN} = -1mA,1KΩ +ON-resistance	9	10	11	V
Lower Output Low voltage	$V_{OUTN(L)}$	IOUTN=1mA, 50 to 100Ω	−	0.05	0.1	V
Current limit detection voltage of Idc	V_{Idc}	−	135	150	165	mV

26.7 AC Electrical Characteristics

26.7.1 AC measurement condition

AC measurement condition

- Output levels: High = 0.8 × VDD / Low = 0.2 × VDD
- Input levels: Refer to low-level input voltage and high-level input voltage in DC Electrical Characteristics.
- Load capacity: CL=30pF

Note: VDD = DVDD5B = AVDD5B

26.7.2 Serial Channel Timing (SIO/UART)

26.7.2.1 I/O Interface mode

In the table below, the letter x represents the period of the system clock (fsys). It varies depending on the programming of the clock gear function.

(1) SCLK input mode

[Data Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	100	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	100	−	
SCLK cycle	t_{SCY}	8x	−	200	−	
Valid Data Input ← SCLK rise or fall (Note1)	t_{SRD}	30	−	30	−	
SCLK rise or fall → Input Data hold (Note 1)	t_{HSR}	x + 30	−	55	−	

[Data Output]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	120 (Note 3)	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	120 (Note 3)	−	
SCLK cycle	t_{SCY}	8x	−	240	−	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	t_{SCY}/2 − 3x− 45	−	0 (Note 2)	−	
SCLK rise or fall → Output Data hold (Note 1)	t_{OHS}	t_{SCY}/2	−	120	−	

Note 1: SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

Note 2: A calculated value should use it the SCLK cycle of the range which is not subtracted.

Note 3: t_{OSS} shows the minimum which is not subtracted.

(2) SCLK Output mode

[Data Input]

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	2x	−	100	−	ns
Output Data ← SCLK rise	t_{OSS}	$t_{SCY}/2 - 30$	−	20	−	
SCLK rise → Output Data hold	t_{OHS}	$t_{SCY}/2 - 30$	−	20	−	
Valid Data Input ← SCLK rise	t_{SRD}	45	−	45	−	
SCLK rise → Input Data hold	t_{HSR}	0	−	0	−	

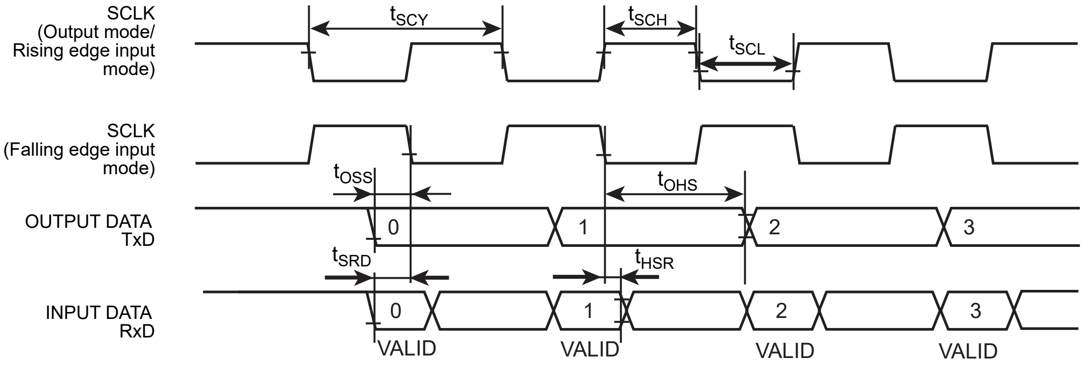


Figure 26-1 Serial channel timing (SIO)

26.7.3 Serial Bus Interface (I2C/SIO)

26.7.3.1 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) = $(2^{n-1} + 14)/x$

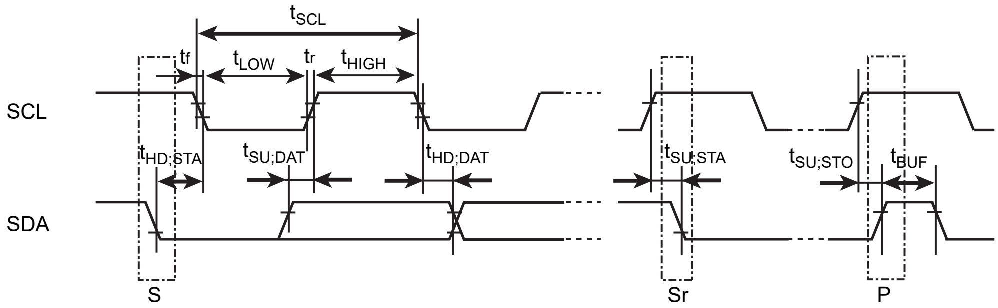
On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400kHz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

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 26-2 Serial Bus timing (I2C)

26.7.4 Event Counter

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2x + 100	−	150	−	ns
Clock high pulse width	t_{VCKH}	2x + 100	−	150	−	ns

26.7.5 Capture

In the table below, the letter x represents the TMRB operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	−	150	−	ns
High pulse width	t_{CPH}	2x + 100	−	150	−	ns

26.7.6 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP release interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT6,7 and C	t_{INTAL}	x + 100	−	125	−	ns
High pulse width for INT6,7, and C	t_{INTAH}	x + 100	−	125	−	ns

2. STOP Release Interrupts

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT6,7 and C	t_{INTBL}	100	−	100	−	ns
High pulse width for INT6,7 and C	t_{INTBH}	100	−	100	−	ns

26.7.7 Debug Communication

26.7.7.1 AC measurement condition

- Output levels : High = 0.7 × DVDD5, Low = 0.3 × DVDD5
- Load capacitance : CL(TRACECLK) = 25pF, CL(TRACEDATA) = 20pF

26.7.7.2 SWD Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	–	ns
DATA hold after CLK rising	T_{d1}	4	–	
DATA valid after CLK rising	T_{d2}	–	45	
DATA valid to CLK rising	T_{ds}	20	–	
DATA hold after CLK falling	T_{dh}	15	–	

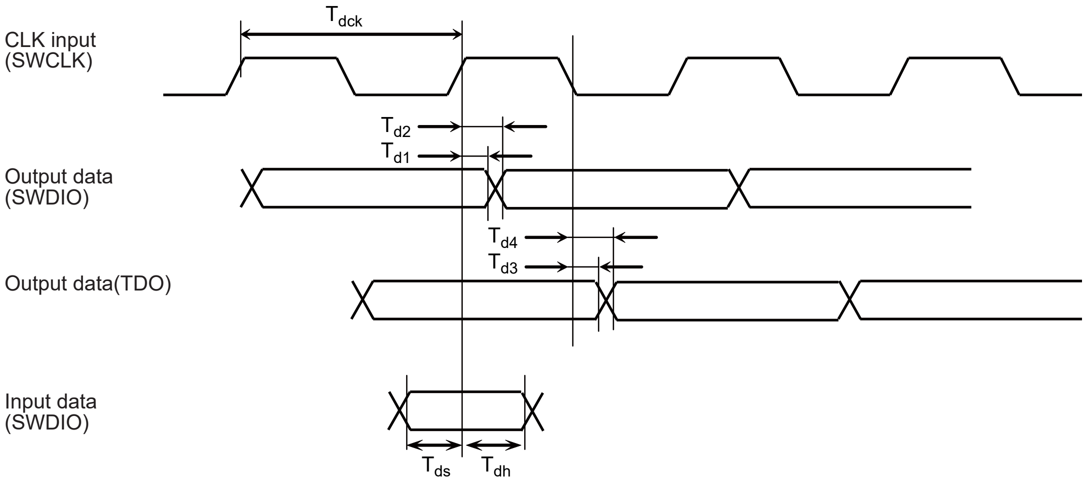


Figure 26-4 SWD communication timing

26.7.8 Flash Characteristics

Parameter	Rating	Min	Typ.	Max	Unit
Guarantee on Flash-memory rewriting	Ta = 0 to 70°C VDD5B = AVDD5B =4.5 to 5.0	−	−	100	times

26.7.9 On chip Oscillator

Parameter	Symbol	Rating	Min	Typ.	Max	Unit
Oscillation frequency	fosc2	Ta = -40 to 85°C	9.4	9.7	10	MHz

Note: Factory default value

26.7.10 External Oscillator

Parameter	Symbol	Rating	PLL	Min	Typ.	Max	Unit
High frequency Oscillation	fosc1	Ta = -40 to 85°C	4	9.9	10	10.1	MHz

26.8 Oscillation Circuit

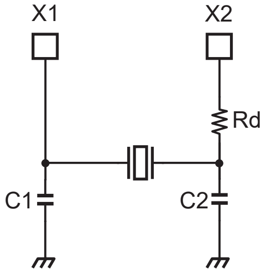


Figure 26-5 High-frequency oscillation connection

Note 1: The load value of the oscillator is the sum of loads (C1 and C2) and the floating load of the actual assembled board. There is a possibility of operating error when using C1 and C2 values in the table below. When designing the board, design the minimum length pattern around the oscillator. We also recommend that oscillator evaluation be carried out using the actual board.

Note 2: Do not be driven X1/X2 by external driver.

The TMP37AFSQG has been evaluated by the oscillator vendor below. Use this information when selecting external parts.

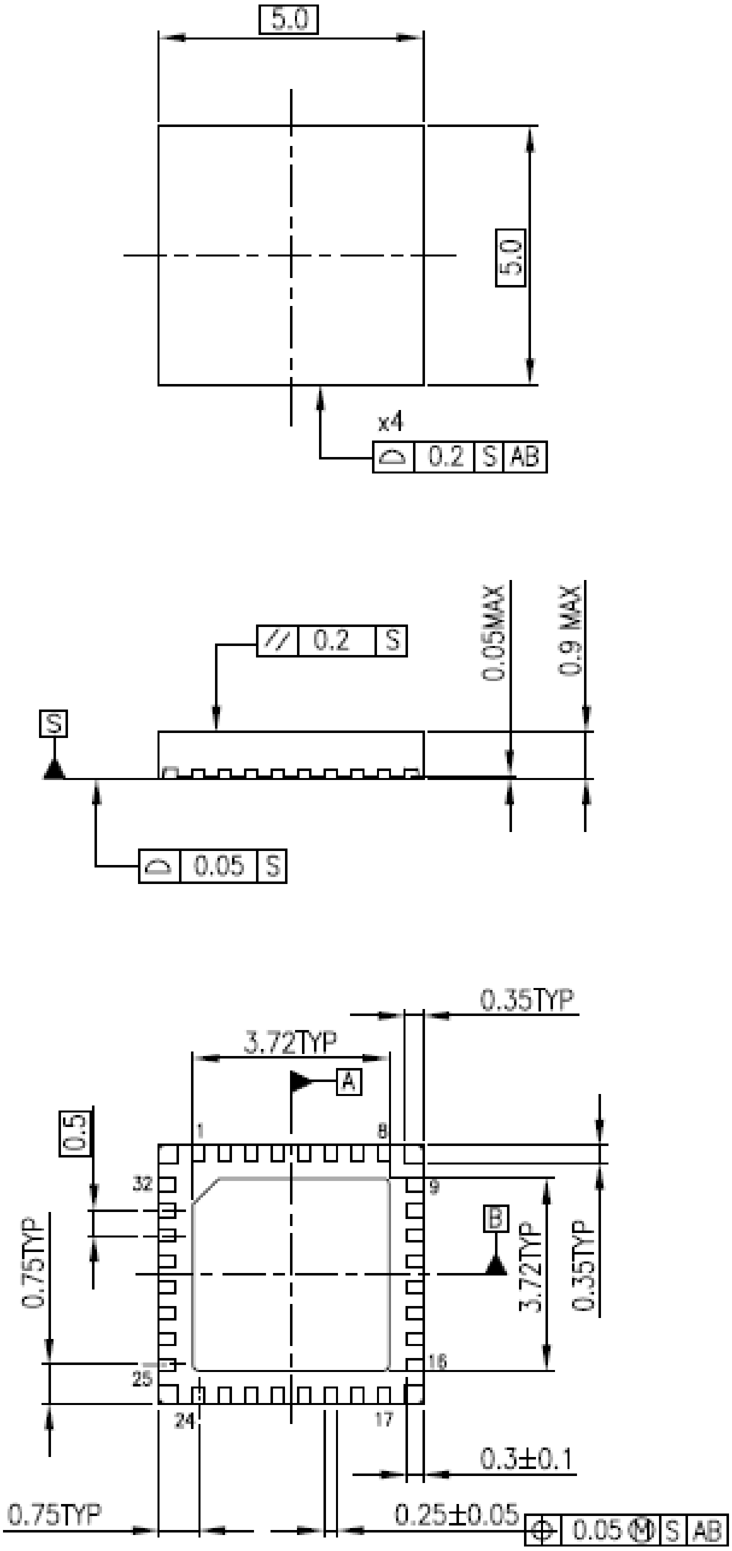
26.8.1 Recommended ceramic oscillator

This product has been evaluated by the ceramic oscillator of Murata Manufacturing Co., Ltd.

Please refer to the Murata Website for details.

27. Package Dimensions

P-VQFN32-0505-0.50-003 "Unit:mm"



IC Usage Considerations

Notes on handling of ICs

[1] The absolute maximum ratings of a semiconductor device are a set of ratings that must not be exceeded, even for a moment. Do not exceed any of these ratings.
Exceeding the rating(s) may cause the device breakdown, damage or deterioration, and may result injury by explosion or combustion.

[2] Use an appropriate power supply fuse to ensure that a large current does not continuously flow in case of over current and/or IC failure. The IC will fully break down when used under conditions that exceed its absolute maximum ratings, when the wiring is routed improperly or when an abnormal pulse noise occurs from the wiring or load, causing a large current to continuously flow and the breakdown can lead smoke or ignition. To minimize the effects of the flow of a large current in case of breakdown, appropriate settings, such as fuse capacity, fusing time and insertion circuit location, are required.

[3] If your design includes an inductive load such as a motor coil, incorporate a protection circuit into the design to prevent device malfunction or breakdown caused by the current resulting from the inrush current at power ON or the negative current resulting from the back electromotive force at power OFF. IC breakdown may cause injury, smoke or ignition.
Use a stable power supply with ICs with built-in protection functions. If the power supply is unstable, the protection function may not operate, causing IC breakdown. IC breakdown may cause injury, smoke or ignition.

[4] Do not insert devices in the wrong orientation or incorrectly.

Make sure that the positive and negative terminals of power supplies are connected properly.
Otherwise, the current or power consumption may exceed the absolute maximum rating, and exceeding the rating(s) may cause the device breakdown, damage or deterioration, and may result injury by explosion or combustion.
In addition, do not use any device that is applied the current with inserting in the wrong orientation or incorrectly even just one time. Utmost care is necessary in the design of board layout since the IC may be destroyed and cause smoke or ignition by short-circuiting between outputs, air contamination faults, or faults due to improper grounding, or by short-circuiting between contiguous pins. Specially, in the design of the output, VM, VINREG5, DVDD5, AVDD5, UH, UL, VH, VL, WH, WL, VREG, VOUT3, VOUT15, DVSS, AVSS and GND lines which have high voltage and high current, utmost care is necessary.

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT. For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

https://toshiba.semicon-storage.com/