TOSHIBA

32 Bit RISC Microcontroller
TX04 Series

TMPM475FDFG/FZFG/FYFG

TOSHIBA CORPORATION
Storage & Electronic Devices Solutions Company

ARM

General precautions on the use of Toshiba MCUs

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

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

1. The MCUs' operation at power-on

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

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

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

2. Unused pins

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

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

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

3. Clock oscillation stability

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

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

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

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

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

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

Base Address = 0x0000_0000

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

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

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

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

1.2.2 SAMCR(Control register)

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

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

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

R / W	READ WRITE
R	READ
W	WRITE

c. Data description

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

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

d. Register 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
2015/01/09	Tentative 1	First Release
2015/05/20	Tentative 2	Contents Revised
-	-	-
2015/09/16	1	First Release
2015/12/24	2.1	Contents Revised
2016/02/18	2.2	Contents Revised
2016/03/10	2.3	Contents Revised
2021/10/15	2.4	Contents Revised
2022/06/01	2.5	Contents Revised
2023/07/14	2.6	Contents Revised
2023/07/31	2.7	Contents Revised

Table of Contents

7. Internal High-speed Oscillation Adjustment Function

8. Exceptions

9. Digital Noise Filter Circuit (DNF)

10. µDMA Controller (µDMAC)

11. Input / Output port

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

14. Serial Bus Interface (I2C/SIO)

15. CAN Controller (CAN)

18. Vector Engine (A-VE)

19. Encoder Input Circuit (A-ENC)

24. Flash Memory (FLASH)

25. Debug Interface

26. Port Section Equivalent Circuit Schematic

27. Electrical Characteristics

28. Package Dimensions

CMOS 32-Bit Microcontroller

TMPM475FDFG/FZFG/FYFG

TMPM475FDFG/FZFG/FYFG is a 32-bit RISC microprocessor series with an ARM®Cortex®-M4F microprocessor core.

Features of the TMPM475FDFG/FZFG/FYFG are as follows:

1.1 Features

1. ARM®Cortex®-M4F microprocessor core
   a. Improved code efficiency has been realized through the use of Thumb®-2 instruction.

      New 16-bit Thumb instructions for improved program flow

      New 32-bit Thumb instructions for improved performance

      New Thumb mixed 16-/32-bit instruction set can produce faster, more efficient code.
   b. Both high performance and low power consumption have been achieved.

      [High performance]

      - A 32-bit multiplication (32 × 32 = 32 bits) and multiply-accumulate operation (32 × 32 = 32 bits) can be executed with one clock.

      - SIMD (Single Instruction Multiple Data) operation can be executed with one clock.

      - A division takes within 2 to 12 cycles.

      [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. Single-precision floating-point execution unit (FPU)
   - Conformity to IEEE754 standard
   - Addition/subtraction/multiplication can be executed with one clock. Multiply-accumulate operation can be executed with 3 clocks.
   - Dedicated data register can perform parallel processing aside from CPU.

3. On Chip program memory and data memory
   - On-chip RAM :

     TMPM475FDFG : 34 Kbyte

     TMPM475FZFG : 34 Kbyte

     TMPM475FYFG : 18 Kbyte
   - On-chip FlashROM :

     TMPM475FDFG : 512 Kbyte

     TMPM475FZFG : 384 Kbyte

TMPM475FYFG : 256 Kbyte

4. µDMA controller (µDMAC) : 32 channels / 1 unit

   Transfer mode : Built-in memory and peripheral function

5. Clock generator (CG)
   - Installed 1 unit of Built-in PLL. (8x, 10x, 12x)
   - Clock gear function : The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8 or 1/16.

6. Low power consumption function

   IDLE, STOP

7. Input/ output ports (PORT) : 79 pins

8. 16-bit timer (TMRB) : 10 channels
   - 16-bit interval timer mode
   - 16-bit event counter mode
   - Input capture function
   - 16-bit PPG output
   - External trigger PPG output

9. Watchdog timer (WDT) : 1 channel

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

10. Serial channel (SIO/UART) : 4 channels
    - Selectable either UART or synchronous mode
    - Transmit FIFO: 4-stage 8-bit width, receive FIFO: 4-stage 8-bit width

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

    Communication rate 100kbps / 400kbps

12. CAN Controller (CAN) : 1 unit
    - Compliant with CAN version 2.0 B (active)
    - 32 Mailboxes
    - CAN bus baud rate up to 1 Mbps

13. Power_On reset function (POR) : 1 unit

14. Voltage detect function (VLTD) : 1 unit

15. Oscillation frequency detect function (OFD) : 1 unit

16. Vector engine (A-VE) : 2 channels
    - Calculation circuit for motor control
    - Corresponding to 2 motor

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

18. Encoder input circuit (A-ENC) : 2 channels
    - Position detection functions

      Support incremental type encoders (AB encoder and ABZ encoder).

      Support Hall sensor ICs (1 to 3 phase inputs).

      Support zero-cross detection for induced voltage of 3-phase BLDC motors in 120-degree conduction.
    - Incorporates the digital-noise filter in the input circuit.
    - Sampling input can be performed synchronizing with PWM signals from the PMD.
    - Timer counter functions

      16-bit up/down counter depending on position detection using encoders and Hall sensor ICs.

      16-bit timer counter/capture count operation can be set at the specified cycle.

      32-bit timer counter/capture

19. 12-bit AD converter (ADC) : 2 units

    Analog input to unit A : 12 channels

    Analog input to unit B : 11 channels

    External analog input pin to unit A : 9 pins

    External analog input pin to unit A and B : 3 pins

    External analog input pin to unit B : 8 pins
    - Start by the internal trigger : TMRB interrupt / PMD trigger
    - Constant conversion mode
    - AD monitoring 2 channels
    - Conversion speed 1.0 μsec (@ADC conversion clock = 120 MHz)

20. Interrupt source : The order of precedence can be set over 7 levels.
    - Internal 80 factors (except the watchdog timer interrupt)
    - External 16 factors

21. Input digital noise filter (DNF) : External interrupt source 16 factors

22. Endian

    Little endian

23. Internal high-speed oscillation circuit : 10 MHz

24. Internal high-speed oscillation adjustment function (TRMOSC) : 1 unit

25. Maximum operating frequency : 120 MHz

26. Operating voltage range

- DVDD5 = 4.5V to 5.5V (fsys=120MHz)

  All function operation
- DVDD5 = 3.9V to 4.5V (fsys=120MHz)

  Restrictions of 12-bit ADC and AC Electrical Characteristics, and Flash writing.

27. Temperature range

    -40 °C to 85 °C

28. Package

    LQFP100 (14mm×14mm, 0.5mm pitch)

1.2 Block Diagram

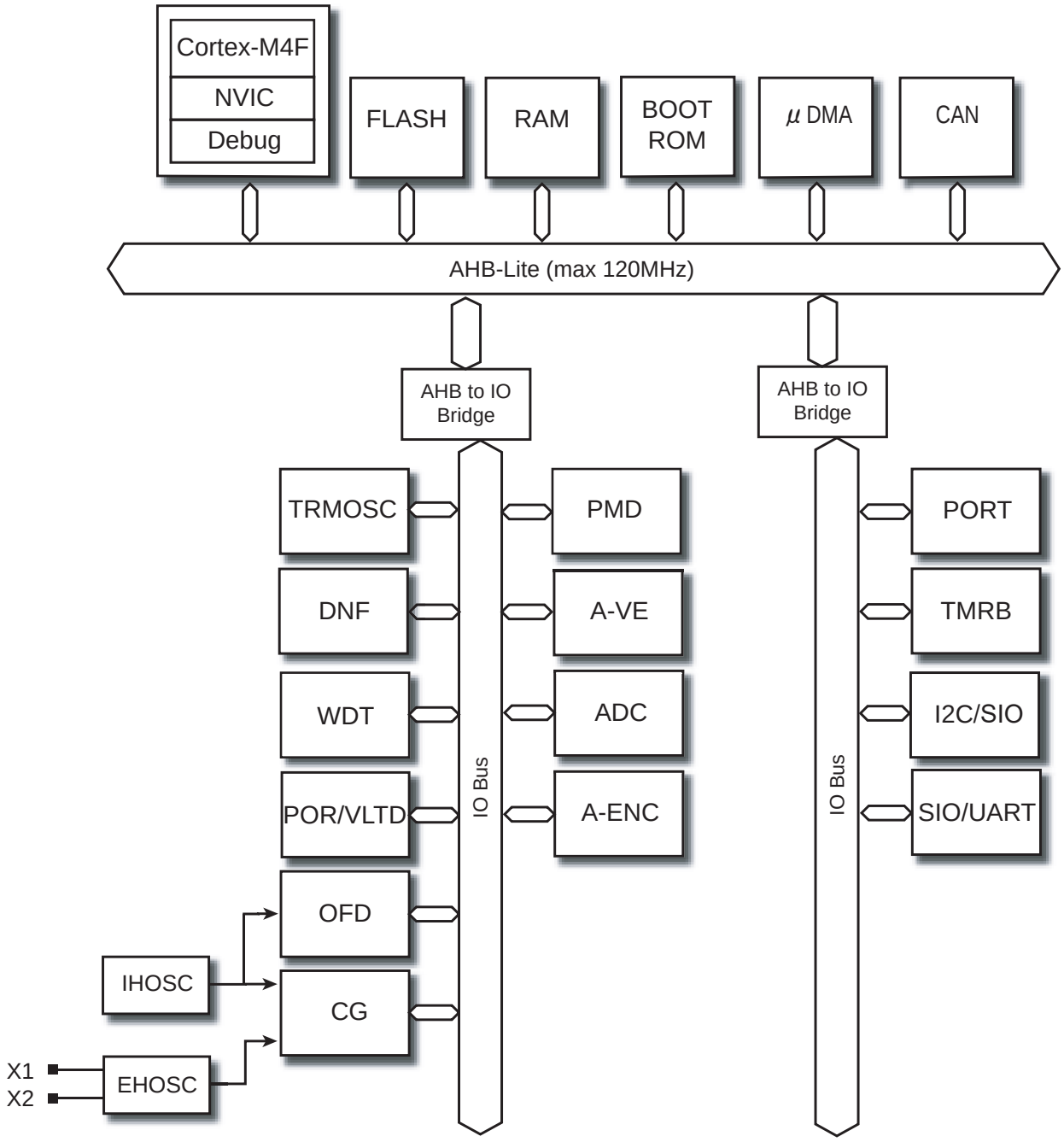


Figure 1-1 TMPM475FDFG/FZFG/FYFG block diagram

1.3 Pin Layout (Top view)

The pin layout of TMPM475FDFG/FZFG/FYFG is a figure below.

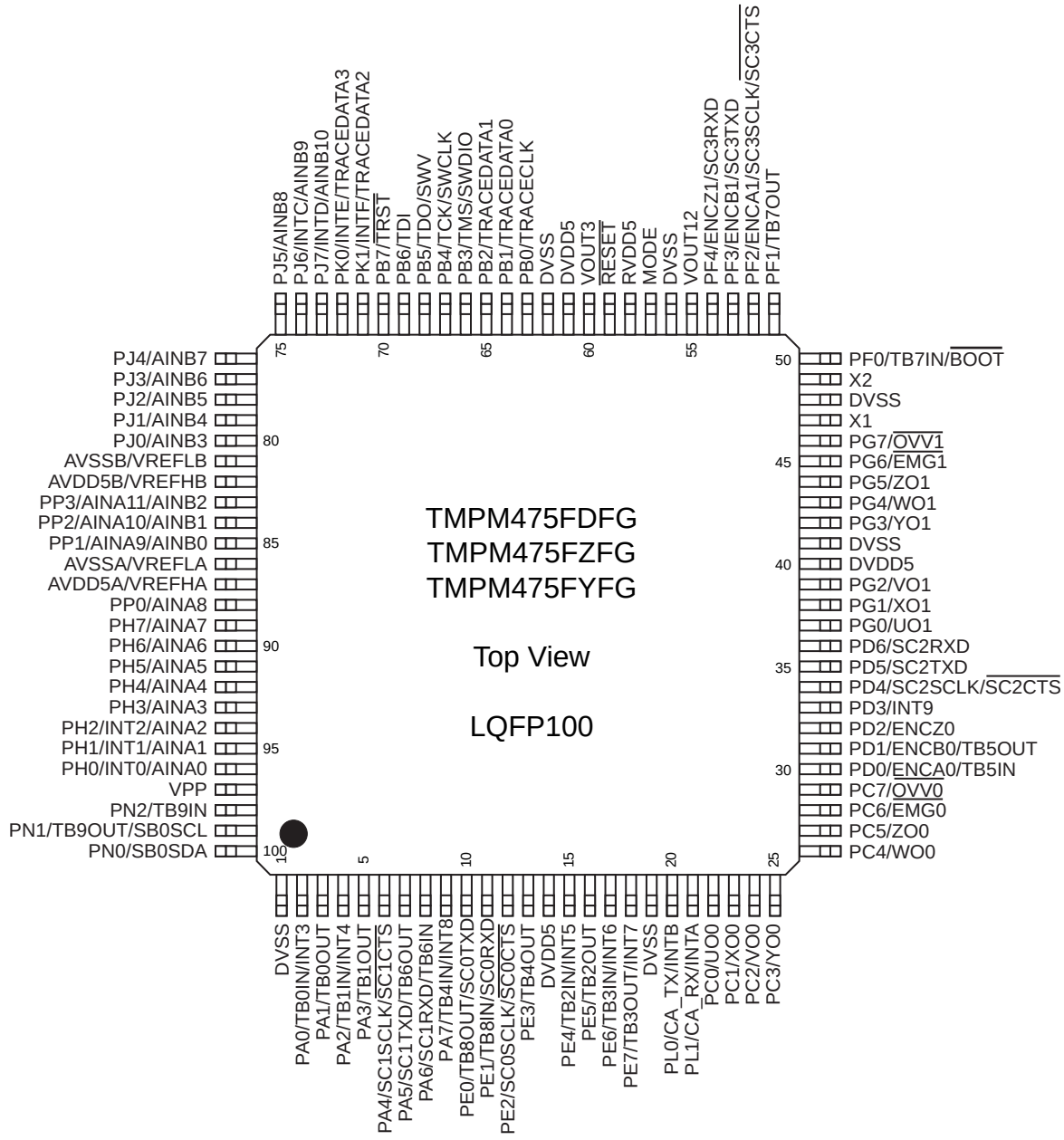


Figure 1-2 Pin layout (LQFP100)

1.4 Pin names and Functions

1.4.1 Pin names and Functions for each peripheral function, control pin and power supply pin

1.4.1.1 Peripheral functions

Table 1-1 Pin names and functions

Peripheral function	Pin name	Input or Output	Function
External interrupt	INTx	Input	External interrupt input pin x External interrupt input pin x has a noise filter (Filter width 30ns typ.).
16-bit timer / even 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	Hand shake input pin
I2C	SBxSDA	I/O	Data input / output pin
	SBxSCL	I/O	Clock input / output pin
CAN	CA_TX	Output	Transmit data output pin
	CA_RX	Input	Receive data input pin
Analog digital convertor	AINAx	Input	Analog input pin
	AINBx	Input	Analog input pin
Encoder	ENCAx	Input	Encoder input pin
	ENCBx	Input	Encoder input pin
	ENCZx	Input	Encoder input pin
PMD	XOx	Output	X-phase output pin
	YOx	Output	Y-phase output pin
	ZOx	Output	Z-phase output pin
	UOx	Output	U-phase output pin
	VOx	Output	V-phase output pin
	WOx	Output	W-phase output pin
	$\overline{\text{EMGx}}$	Input	Emergency status detection input
	$\overline{\text{OVVx}}$	Input	Overvoltage detection input

1.4.1.2 Debug functions

Table 1-2 Pin names and functions

Pin name	Input or Output	Function
TMS	Input	JTAG test mode select input pin
TCK	Input	JTAG serial data input pin
TDO	Output	JTAG serial data output pin
TDI	Input	JTAG test reset input pin
$\overline{\text{TRST}}$	Input	Serial wire reset input pin
SWDIO	I/O	Serial wire data input / output pin
SWCLK	Input	Serial wire clock input pin
SWV	Output	Serial wire viewer output pin
TRACECLK	Output	Trace clock output pin
TRACEDATA0	Output	Trace data output pin 0
TRACEDATA1	Output	Trace data output pin 1
TRACEDATA2	Output	Trace data output pin 2
TRACEDATA3	Output	Trace data output pin 3

1.4.1.3 Control functions

Table 1-3 Pin names and functions

Pin name	Input or Output	Function
X1	Input	High frequency resonator connection pin
X2	Output	High frequency resonator connection pin
MODE	Input	MODE pin This pin must be fixed to Low level.
$\overline{\text{RESET}}$	Input	Reset signal input pin
$\overline{\text{BOOT}}$	Input	BOOT mode control pin BOOT mode control pin is sampled at the rising edge of reset signal input pin. TMPM475FDFG/FZFG/FYFG changes Single Boot Mode when BOOT mode control pin is "Low". TMPM475FDFG/FZFG/FYFG changes Single Chip Mode when BOOT mode control pin is "High". Refer to "Flash memory" section for a detail.

1.4.1.4 Power supply pins

Table 1-4 Pin names and functions

Power supply pin name	Function
VOUT12	Pin connected with the capacitor (3.3 to 4.7μF) for the regulator
VOUT3	Pin connected with the capacitor (3.3 to 4.7μF) for the regulator
VPP	(This pin must be open.)
RVDD5	Power supply pin for the regulator
DVDD5	Power supply pin for the digital circuit DVDD5 supplies the following pins. PA, PB, PC, PD, PE, PF, PG, PL, PK, PN, X1, X2, MODE, $\overline{\text{RESET}}$
DVSS	GND pin for the digital circuit
AVDD5A AVDD5B	Power supply pin for the analog circuit (Note 1) AVDD5A supplies the following pins. PH, PP0 AVDD5B supplies the following pins. PJ, PP1 to 3
AVSSA AVSSB	GND pin for ADC (Note 2)
VREFHA VREFHB	Reference power supply pin for ADC
VREFLA VREFLB	Reference power supply pin for ADC

Note 1: Must be connected to power supply even if AD converter is not used.
Note 2: AVSSB must be connected to GND even if the AD converter is not used.

1.4.1.5 Capacitors between power supply pins

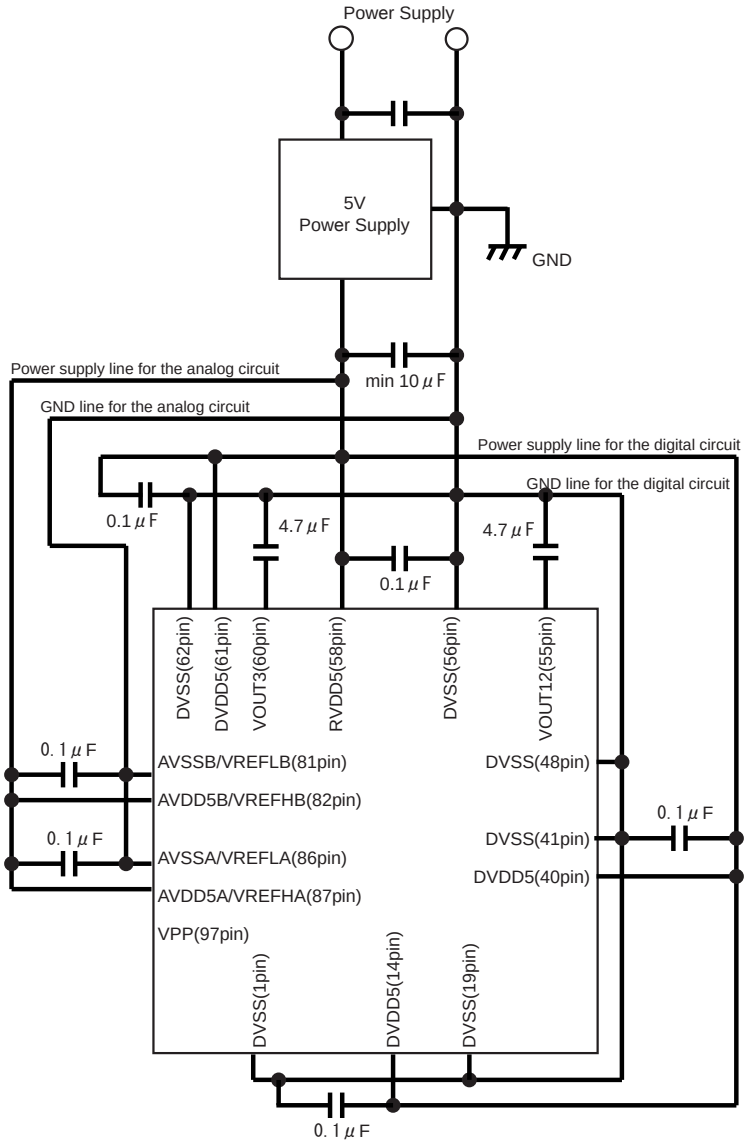


Figure 1-3 Capacitors for power supply pins connection circuit

Note 1: 5V power supply output capacitor (min 10 μF) must be place on the shortest distance from the 58 pin RVDD5, 56 pin DVSS.

Note 2: 5V power supply and GND lines must be separated for the analog circuit and the digital circuit in near 5V power supply output capacitor (min 10 μ F) between this capacitor and 58 pin RVDD5, 56 pin DVSS.

Note 3: Power supply and GND lines must be bring close and be wired. When they are away, a power supply loop make for power supply and GND lines through the capacitor of a power supply circuit, and they will be the antenna received high frequency noise.

Note 4: Capacitors of VOUT3 and VOUT12 for regulators must be same capacity, and they must be place on the shortest distance from 56 pin DVSS.

Note 5: Capacitors between 14 pin and 19 pin, 40 pin and 41 pin, 58 pin and 56 pin, 61 pin and 62 pin must be place on the shortest distance from the pins.

Note 6: Capacitors between 82 pin and 81 pin, 87 pin and 86 pin must be place on the shortest distance from the pins.

1.4.2 Pin names and Functions of TMPM475FDFG/FZFG/FYFG

1.4.2.1 The detail for pin names and function list

The mean of the symbol in the table is shown below.

1. Function A

   The function which is specified without setting of function register is shown in this cell.

2. Function B

   The function which is specified with setting of function register is shown in this cell. The number in this cell is corresponded with the number of function register.

3. Pin specification

   The mean of the symbol in the table is shown below.

   - SMT/CMOS : Type of input gate

     SMT : Schmitt input

     CMOS : CMOS input

   - 5V_T : 5V tolerant support

     Yes : supported

     N/A : Not supported

   - OD : Programmable open drain output support

     Yes : supported

     N/A : Not supported

   - PU/PD:Programmable Pull-Up / Pull-Down support

     PU : Programmable Pull-Up supported

     PD : Programmable Pull-Down supported

1.4.2.2 PORT / Debug pins

Table 1-5 Pin names and functions <Sorted by PORT>

Pin No.	PORT	Function A	Function B					Port Specification		
			1	2	3	4	5	PU/PD	OD	SMT/ CMOS
PORT A										
2	PA0	INT3	TB0IN					PU/PD	Yes	SMT
3	PA1		TB0OUT					PU/PD	Yes	SMT
4	PA2	INT4	TB1IN					PU/PD	Yes	SMT
5	PA3		TB1OUT					PU/PD	Yes	SMT
6	PA4		SC1SCLK	$\overline{\text{SC1CTST}}$				PU/PD	Yes	SMT
7	PA5		SC1TXD	TB6OUT				PU/PD	Yes	SMT
8	PA6		SC1RXD	TB6IN				PU/PD	Yes	SMT
9	PA7	INT8	TB4IN					PU/PD	Yes	SMT
PORTB										
63	PB0		TRACECLK					PU/PD	Yes	SMT
64	PB1		TRACEDATA0					PU/PD	Yes	SMT
65	PB2		TRACEDATA1					PU/PD	Yes	SMT
66	PB3		TMS/SWDIO					PU/PD	Yes	SMT
67	PB4		TCKSWCLK					PU/PD	Yes	SMT
68	PB5		TDO/SWV					PU/PD	Yes	SMT
69	PB6		TDI					PU/PD	Yes	SMT
70	PB7		$\overline{\text{TRST}}$					PU/PD	Yes	SMT
PORTC										
22	PC0		UO0					PU/PD	Yes	SMT
23	PC1		XO0					PU/PD	Yes	SMT
24	PC2		VO0					PU/PD	Yes	SMT
25	PC3		YO0					PU/PD	Yes	SMT
26	PC4		WO0					PU/PD	Yes	SMT
27	PC5		ZO0					PU/PD	Yes	SMT
28	PC6		$\overline{\text{EMG0}}$					PU/PD	Yes	SMT
29	PC7		$\overline{\text{OVV0}}$					PU/PD	Yes	SMT
PORTD										
30	PD0		ENCA0	TB5IN				PU/PD	Yes	SMT
31	PD1		ENCB0	TB5OUT				PU/PD	Yes	SMT
32	PD2		ENCZ0					PU/PD	Yes	SMT
33	PD3	INT9						PU/PD	Yes	SMT
34	PD4		SC2SCLK	$\overline{\text{SC2CTS}}$				PU/PD	Yes	SMT
35	PD5		SC2TXD					PU/PD	Yes	SMT
36	PD6		SC2RXD					PU/PD	Yes	SMT
PORTE										
10	PE0		SC0TXD	TB8OUT				PU/PD	Yes	SMT
11	PE1		SC0RXD	TB8IN				PU/PD	Yes	SMT
12	PE2		SC0SCLK	$\overline{\text{SC0CTS}}$				PU/PD	Yes	SMT
13	PE3		TB4OUT					PU/PD	Yes	SMT
15	PE4	INT5	TB2IN					PU/PD	Yes	SMT
16	PE5		TB2OUT					PU/PD	Yes	SMT

Table 1-5 Pin names and functions <Sorted by PORT>

Pin No.	PORT	Function A	Function B					Port Specification		
			1	2	3	4	5	PU/PD	OD	SMT/ CMOS
17	PE6	INT6	TB3IN					PU/PD	Yes	SMT
18	PE7	INT7	TB3OUT					PU/PD	Yes	SMT
PORTF										
50	PF0	$\overline{\text{BOOT}}$	TB7IN					PU/PD	Yes	SMT
51	PF1		TB7OUT					PU/PD	Yes	SMT
52	PF2		ENCA1	SC3SCLK	$\overline{\text{SC3CTS}}$			PU/PD	Yes	SMT
53	PF3		ENCB1	SC3TXD				PU/PD	Yes	SMT
54	PF4		ENCZ1	SC3RXD				PU/PD	Yes	SMT
PORTG										
37	PG0		UO1					PU/PD	Yes	SMT
38	PG1		XO1					PU/PD	Yes	SMT
39	PG2		VO1					PU/PD	Yes	SMT
42	PG3		YO1					PU/PD	Yes	SMT
43	PG4		WO1					PU/PD	Yes	SMT
44	PG5		ZO1					PU/PD	Yes	SMT
45	PG6		$\overline{\text{EMG1}}$					PU/PD	Yes	SMT
46	PG7		$\overline{\text{OVV1}}$					PU/PD	Yes	SMT
PORTH										
96	PH0	AINA0 INT0						PU/PD	Yes	SMT
95	PH1	AINA1 INT1						PU/PD	Yes	SMT
94	PH2	AINA2 INT2						PU/PD	Yes	SMT
93	PH3	AINA3						PU/PD	Yes	SMT
92	PH4	AINA4						PU/PD	Yes	SMT
91	PH5	AINA5						PU/PD	Yes	SMT
90	PH6	AINA6						PU/PD	Yes	SMT
89	PH7	AINA7						PU/PD	Yes	SMT
PORTJ										
80	PJ0	AINB3						PU/PD	Yes	SMT
79	PJ1	AINB4						PU/PD	Yes	SMT
78	PJ2	AINB5						PU/PD	Yes	SMT
77	PJ3	AINB6						PU/PD	Yes	SMT
76	PJ4	AINB7						PU/PD	Yes	SMT
75	PJ5	AINB8						PU/PD	Yes	SMT
74	PJ6	AINB9 INTC						PU/PD	Yes	SMT
73	PJ7	AINB10 INTD						PU/PD	Yes	SMT
PORTK										
72	PK0	INTE	TRACEDATA3					PU/PD	Yes	SMT
71	PK1	INTF	TRACEDATA2					PU/PD	Yes	SMT
PORTL										
20	PL0	INTB	CA_TX					PU/PD	Yes	SMT
21	PL1	INTA	CA_RX					PU/PD	Yes	SMT

Table 1-5 Pin names and functions <Sorted by PORT>

Pin No.	PORT	Function A	Function B					Port Specification		
			1	2	3	4	5	PU/PD	OD	SMT/ CMOS
PORTN										
100	PN0			SB0SDA				PU/PD	Yes	SMT
99	PN1		TB9OUT	SB0SCL				PU/PD	Yes	SMT
98	PN2		TB9IN					PU/PD	Yes	SMT
PORTP										
88	PP0	AINA8						PU/PD	Yes	SMT
85	PP1	AINA9 AINB0						PU/PD	Yes	SMT
84	PP2	AINA10 AINB1						PU/PD	Yes	SMT
83	PP3	AINA11 AINB2						PU/PD	Yes	SMT

1.4.2.3 Control pins

Table 1-6 The number of pin and pin names

Pin No.	Control function pin name
47	X1
49	X2
57	MODE
59	$\overline{\text{RESET}}$
50	$\overline{\text{BOOT}}$

1.4.2.4 Power Supply pins

Table 1-7 The number of pin and pin names

Pin No.	Power supply pin name
55	VOUT12
60	VOUT3
58	RVDD5
97	VPP
14, 40, 61	DVDD5
1, 19, 41, 48, 56, 62	DVSS
87	AVDD5A VREFHA
82	AVDD5B VREFHB
86	VREFLB AVSSB
81	VREFLB AVSSB

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 DMA Controller (DMAC)"
- "2.1.2 16-bit Timer / Event Counter (TMRB)"
- "2.1.3 Serial Channel (SIO/UART)"
- "2.1.4 Serial Bus Interface (I2C)"
- "2.1.5 CAN Controller (CAN)"
- "2.1.6 Vector Engine (A-VE)"
- "2.1.7 Motor Control Circuit (PMD : Programmable Motor Driver)"
- "2.1.8 Encoder Input Circuit (A-ENC)"
- "2.1.9 Analog/Digital Converter (ADC)"
- "2.1.10 Watchdog Timer(WDT)"
- "2.1.11 Debug Interface"

2.1 Information of Each Peripheral Function

2.1.1 DMA Controller (DMAC)

TMPM475FDFG/FZFG/FYFG incorporates 1 unit of built-in DMA controller.

Table 2-1 DMA Request Table

Channel	Burst	Single
0	SIO/UART0 reception	−
1	SIO/UART0 transmission	−
2	SIO/UART1 reception	−
3	SIO/UART1 transmission	−
4	SIO/UART2 reception	−
5	SIO/UART2 transmission	−
6	SIO/UART3 reception	−
7	SIO/UART3 transmission	−
8	AD conversion A triggered by timer is finished	−
9	AD conversion B triggered by timer is finished	−
10	AD conversion A started by software is finished	−
11	AD conversion B started by software is finished	−
12	AD conversion A triggered by PMD0 is finished	−
13	AD conversion B triggered by PMD0 is finished	−
14	AD conversion A triggered by PMD1 is finished	−
15	AD conversion B triggered by PMD1 is finished	−
16	PMD0 PWM interrupt	−
17	PMD1 PWM interrupt	−
18	−	−
19	−	−
20	TMRB0 input capture 0	−
21	TMRB1 input capture 0	−
22	TMRB2 input capture 0	−
23	TMRB3 input capture 0	−
24	VE interrupt 0	−
25	VE interrupt 1	−
26	TMRB0 compare match (Note)	−
27	TMRB1 compare match (Note)	−
28	TMRB2 compare match (Note)	−
29	TMRB3 compare match (Note)	−
30	VE0 interrupt by task is finished	−
31	VE1 interrupt by task is finished	−

Note: A DMA transfer request of TMRB occurs under the same conditions as a TMRB interrupt. A TMRB interrupt occurs when the up-counter matches timer register 0/1, or the up-counter overflows. Mask unnecessary factors with interrupt mask register (TBxIM) if required.

2.1.2 16-bit Timer / Event Counter (TMRB)

TMPM475FDFG/FZFG/FYFG incorporates 10 channels of TMRB.

Table 2-2 Pin Specifications

Channel	TBxOUT	TBxIN
TMRB0	PA1	PA0
TMRB1	PA3	PA2
TMRB2	PE5	PE4
TMRB3	PE7	PE6
TMRB4	PE3	PA7
TMRB5	PD1	PD0
TMRB6	PA5	PA6
TMRB7	PF1	PF0
TMRB8	PE0	PE1
TMRB9	PN1	PN2

Table 2-3 Interrupts

Channel	Input capture	Compare match detection / Over flow
TMRB0	INTCAP00, INTCAP01	INTTB00, INTTB01
TMRB1	INTCAP10, INTCAP11	INTTB10, INTTB11
TMRB2	INTCAP20, INTCAP21	INTTB20, INTTB21
TMRB3	INTCAP30, INTCAP31	INTTB30, INTTB31
TMRB4	INTCAP40, INTCAP41	INTTB40, INTTB41
TMRB5	INTCAP50, INTCAP51	INTTB50, INTTB51
TMRB6	INTCAP60, INTCAP61	INTTB60, INTTB61
TMRB7	INTCAP70, INTCAP71	INTTB70, INTTB71
TMRB8	INTCAP80, INTCAP81	INTTB80, INTTB81
TMRB9	INTCAP90, INTCAP91	INTTB90, INTTB91

Table 2-4 Synchronous start specifications

Master channel	Slave channel
TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6
TMRB7	TMRB8, TMRB9

Table 2-5 Capture trigger specifications

Trigger input channel	Trigger output
TMRB0	ENC0 divided pulse
TMRB1	ENC1 divided pulse

2.1.3 Serial Channel (SIO/UART)

TMPM475FDFG/FZFG/FYFG incorporates 4 channels of SIO/UART.

Table 2-6 Pin Specifications

Channel	SCxTXD	SCxRXD	SCxSCLK	$\overline{\text{SCxCTS}}$
SC0	PE0	PE1	PE2	PE2
SC1	PA5	PA6	PA4	PA4
SC2	PD5	PD6	PD4	PD4
SC3	PF3	PF4	PF2	PF2

Table 2-7 Interrupts

Channel	Serial reception	Serial transmission
SC0	INTRX0	INTTX0
SC1	INTRX1	INTTX1
SC2	INTRX2	INTTX2
SC3	INTRX3	INTTX3

Table 2-8 Internal connection specifications

Channel	UART source clock
SC0	TMRB4
SC1	TMRB4
SC2	TMRB7
SC3	TMRB7

2.1.4 Serial Bus Interface (I2C)

TMPM475FDFG/FZFG/FYFG incorporate 1 channel of I2C and SIO is not supported.

Table 2-9 Pin Specifications

Channel	SBxSDA	SBxSCL
I2C0	PN0	PN1

Table 2-10 Interrupt

Channel	Interrupt
I2C0	INTSBI0

2.1.5 CAN Controller (CAN)

TMPM475FDFG/FZFG/FYFG incorporates 1 unit of built-in CAN.

Table 2-11 Pin Specifications

Unit	CA_TX	CA_RX
CAN	PL0	PL1

Table 2-12 Interrupts

Unit	receive completion interrupt	transmit completion interrupt	global Interrupt
CAN	INTCANRX	INTCANTX	INTCANGB

2.1.6 Vector Engine (A-VE)

TMPM475FDFG/FZFG/FYFG incorporates 2 channels of built-in A-VE.

Table 2-13 Pin Specifications

Channel	Interrupt
VE0	INTVCN0
VE1	INTVCN1

Table 2-14 Internal connection specifications

Channel	PWM interrupt signal input	ADC conversion finished signal input	ADC conversion result inputs
VE0	INTPMD0	INTADAPDA INTADBPDA	ADAREG0 to 3
VE1	INTPMD1	INTADAPDB INTADBPDB	ADBREG0 to 3

2.1.7 Motor Control Circuit (PMD : Programmable Motor Driver)

TMPM475FDFG/FZFG/FYFG incorporates 2 channels of PMD.

Table 2-15 Pin Specifications

Channel	$\overline{\text{OVVx}}$ Overvoltage 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
PMD0	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PMD1	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0

Table 2-16 Interrupts

Channel	OVV interrupt	EMG interrupt	PWM interrupt
PMD0	INTOVV0	INTEMG0	INTPMD0
PMD1	INTOVV1	INTEMG1	INTPMD1

Table 2-17 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)
PMD0	VEEMGRS0	VECMPU0, V0, W0	VEOUTCR0	VETRGCMP00,01	VETRGSEL0	ADCA monitoring function compare 0, 1 output
PMD1	VEEMGRS1	VECMPU1, V1, W1	VEOUTCR1	VETRGCMP10,11	VETRGSEL1	ADCB monitoring function compare 0, 1 output

Table 2-18 Internal connection specifications (2/2)

Channel	MDOUT transfer timing signal input
PMD0	INTENC0, INTTB00, CTRGO(ENC0)
PMD1	INTENC1, INTTB10, CTRGO(ENC1)

2.1.8 Encoder Input Circuit (A-ENC)

TMPM475FDFG/FZFG/FYFG incorporates 2 channels of A-ENC.

Table 2-19 Pin Specifications

Channel	ENCAx	ENCBx	ENCZx
ENC0	PD0	PD1	PD2
ENC1	PF2	PF3	PF4

Table 2-20 Interrupts

Channel	Interrupt
ENC0	INTENC0
ENC1	INTENC1

Table 2-21 Internal connection specifications

Channel	Timer pulse input	PWM signal input
ENC0	TB0OUT	PWMON0
ENC1	TB1OUT	PWMON1

2.1.9 Analog/Digital Converter (ADC)

TMPM475FDFG/FZFG/FYFG incorporates 2units of ADC (a 12-bit successive-approximation analog-to-digital converter).

Table 2-22 Pin Specifications

Channel	AINA0 to 7	AINA8	AINA9 to 11 AINB0 to 2	AINB3 to 10
ADCA	PH0 to 7	PP0	PP1 to 3	-
ADCB	-	-	PP1 to 3	PJ0 to 7

Table 2-23 Interrupts

Unit	AD conversion triggered by PMD is finished	AD conversion triggered by timer is finished	AD conversion started by software is finished	AD conversion monitoring function
ADCA	INTADAPDB	INTADATMR	INTADASFT	INTADACPA INTADACPB
ADCB	INTADBPDB	INTADBTMR	INTADBSFT	INTADBCPA INTADBCPB

Table 2-24 Internal connection specifications

Unit	PMD trigger input	TMRTRG
ADCA	PMD0TRG0 to 5	INTTB50
ADCB	PMD1TRG0 to 5	INTTB60

2.1.10 Watchdog Timer(WDT)

In TMPM475FDFG/FZFG/FYFG, WDMOD(Watchdog Timer Mode Register) bit 2 <I2WDT> is not supported, and so write " 0 ".

2.1.11 Debug Interface

TMPM475FDFG/FZFG/FYFG supports serial wire debug ports, JTAG debug ports and trace outputs.

Table 2-25 Pin Specifications

	TMS SWDIO	TCK SWCLK	TDO SWV	TDI	$\overline{\text{TRST}}$
JTAG Serial wire	PB3	PB4	PB5	PB6	PB7

	TRACECLK	TRACEDATA0	TRACEDATA1	TRACEDATA2	TRACEDATA3
Trace output	PB0	PB1	PB2	PK1	PK0

3. Processor Core

The TX04 series has a high-performance 32-bit processor core (the ARM Cortex-M4F processor core). For information on the operations of this processor core, please refer to the documentation set issued by ARM Limited.This chapter describes the functions unique to the TX04 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 TMPM475FDFG/FZFG/FYFG.

Refer to the detailed information about the CPU core and architecture, refer to the ARM documentation set for "the Cortex-M4 series processors" in the following URL:

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

Product Name	Core Revision
TMPM475FDFG/FZFG/FYFG	r0p1

3.2 Configurable Options

The Cortex-M4F core has optional blocks.

The following tables shows the configurable options in the TMPM475FDFG/FZFG/FYFG.

Configurable options	Implementation
MPU (Memory Protection Unit)	Absent
FPB (Flash Patch and Breakpoint)	Two literal comparators Six instruction comparators
DWT (Data Watchpoint and Trace)	Four comparators
ITM (Instrumentation Trace Macrocell)	Present
ETM (Embedded Trace Macrocell)	Present
AHB-AP (AHB Access Port)	Present
HTM Interface (AHB Trace Macrocell Interface)	Absent
TPIU (Trace Port Interface Unit)	Present
WIC (Wake-up Interrupt Controller)	Absent
Debug Port (Serial-Wire or JTAG Debug Port)	Present
FPU (Floating Point Unit)	Present
Bit banding	Present
Constant AHB control	Disable

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-M4F core.

TMPM475FDFG/FZFG/FYFG has 96 interrupt inputs. The number of interrupt inputs is reflected in <INTLINESNUM[3:0]> bit of NVIC register. In this product, if read <INTLINESNUM[3:0]> bit, "0x02" is read out.

3.3.2 Number of Priority Level Interrupt Bits

The Cortex-M4F core can optionally configure the number of priority level interrupt bits from 3 bits to 8 bits.

TMPM475FDFG/FZFG/FYFG 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-M4F 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-M4F core outputs SYSRESETREQ signal when <SYSRESETREQ> bit of Application Interrupt and Reset Control Register are set.

TMPM475FDFG/FZFG/FYFG provides as same as warm reset operation when SYSRESETREQ signal are output.

3.3.5 LOCKUP

When irreparable exception generates, the Cortex-M4F core outputs LOCKUP signal to show a serious error included in software.

TMPM475FDFG/FZFG/FYFG 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-M4F core provides auxiliary fault status registers to supply additional system fault information to software.

However, TMPM475FDFG/FZFG/FYFG is not defined this function. If auxiliary fault status register is read, always "0x0000_0000" is read out.

3.4 Events

The Cortex-M4F 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.

TMPM475FDFG/FZFG/FYFG 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-M4F 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.

TMPM475FDFG/FZFG/FYFG 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-M4F core, the DCode bus system supports exclusive access. However TMPM475FDFG/FZFG/FYFG does not use this function.

3.7 Floating Point Unit (FPU)

This product implements the Cortex-M4F FPU that is the single precision variant of the ARMv7-M Floating-Point Extension (FPv4-SP). It provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard.

The FPU shares address bus and data bus with Cortex-M4F core, and operates in cooperation. It performs add, subtract, and multiply in 1 clock, and multiply and accumulate in 3 clocks. The parallel processing that is different from CPU is possible with using the exclusive data register.

The FPU supports all single-precision data-processing instructions and data types described in the ARM Architecture Reference Manual.

4. Memory Map

4.1 Memory Map

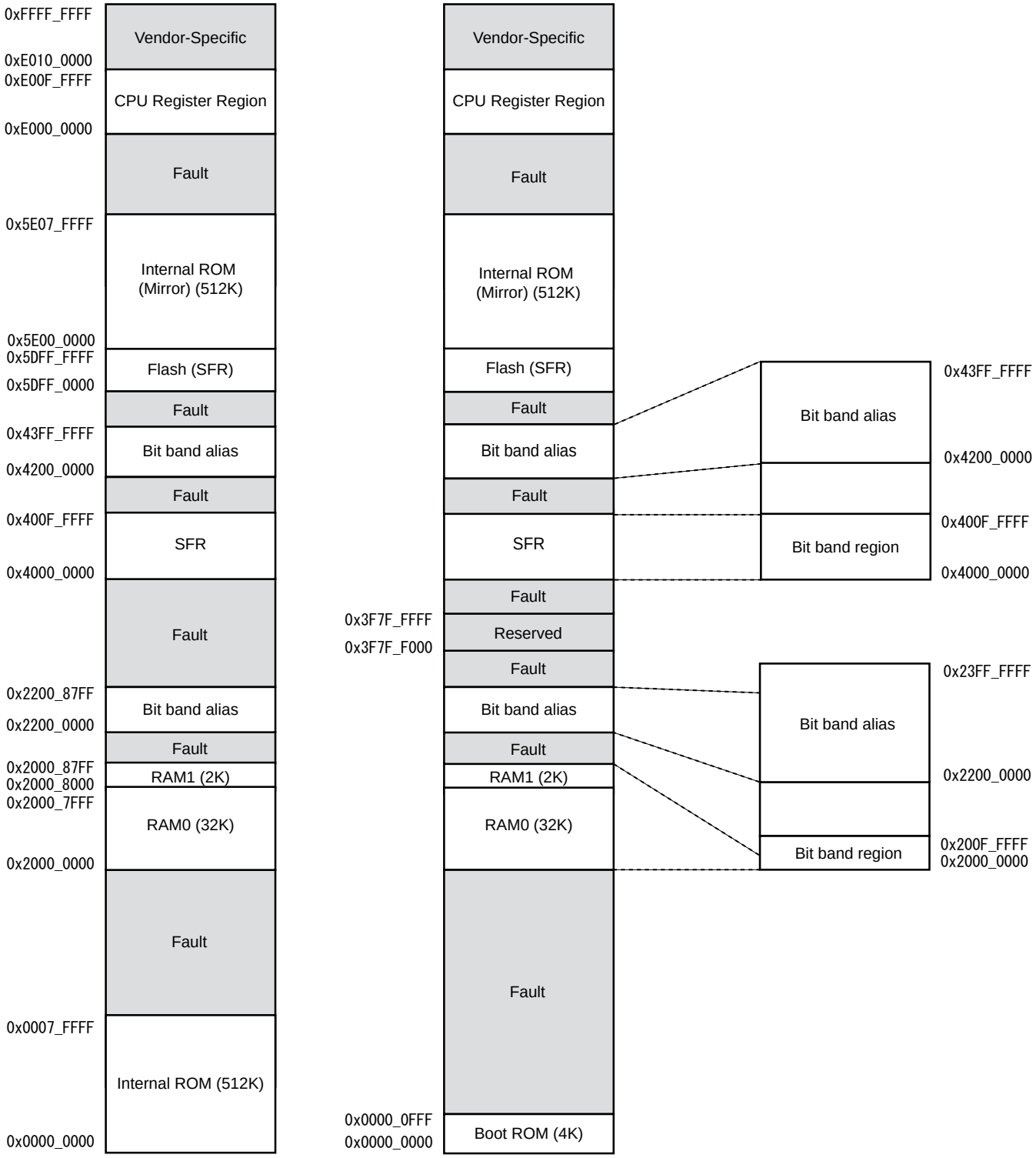
The memory maps for TMPM475FDFG/FZFG/FYFG are based on the ARM Cortex-M4F processor core memory map. The internal ROM, internal RAM and special function registers (SFR) of TMPM475FDFG/FZFG/FYFG are mapped to the Code, SRAM and peripheral regions of the Cortex-M4F respectively. The special function register (SFR) means the control registers of all input/output ports and peripheral functions.

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

For more information on each region, see the "ARM documentation set for the ARM Cortex-M4".

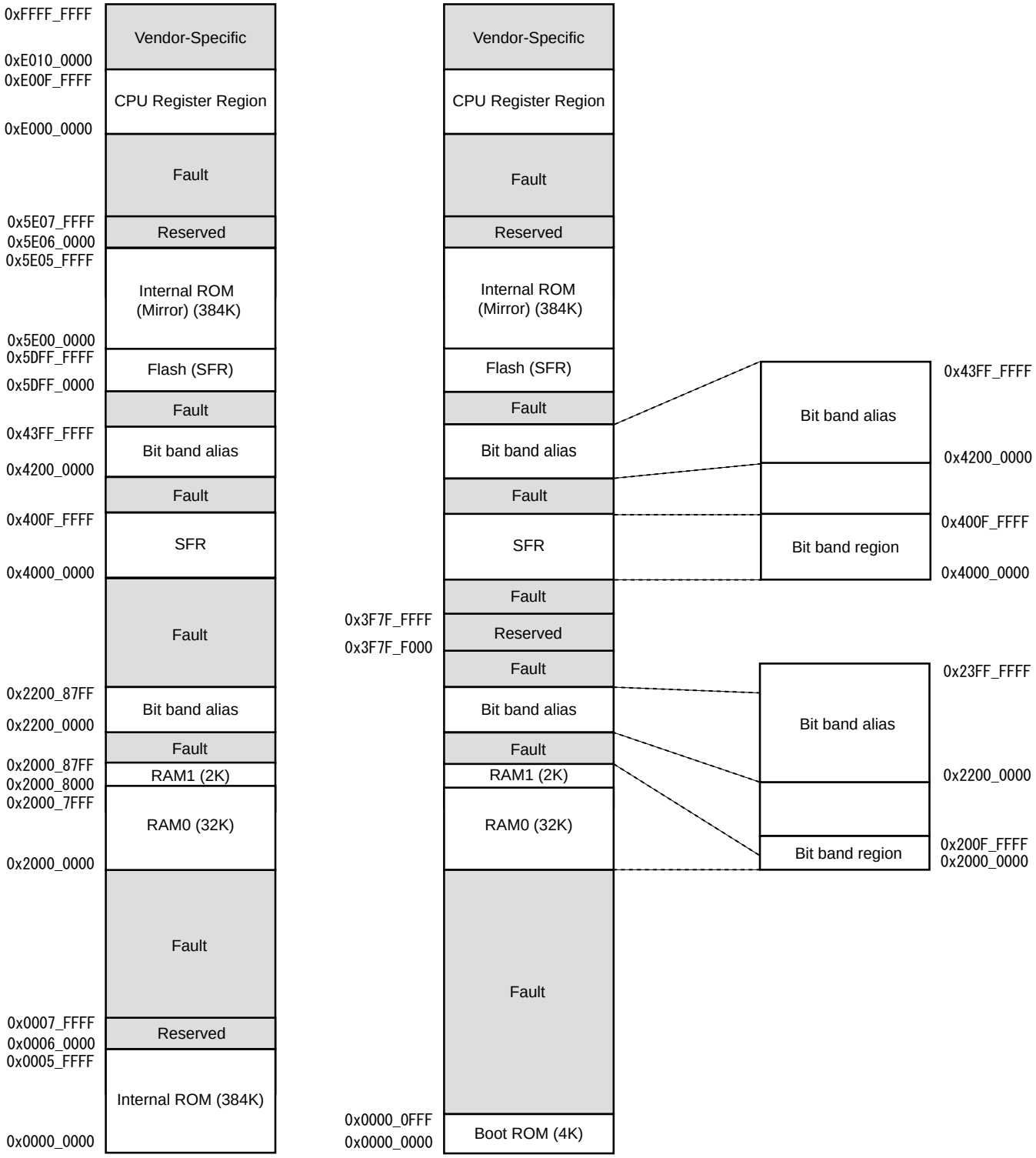
Note that access to regions indicated as "Fault" causes a memory fault if memory faults are enabled, or causes a hard fault if memory faults are disabled. Also, do not access the vendor-specific region.

A memory map of TMPM475FDFG/FZFG/FYFG is shown bellows.



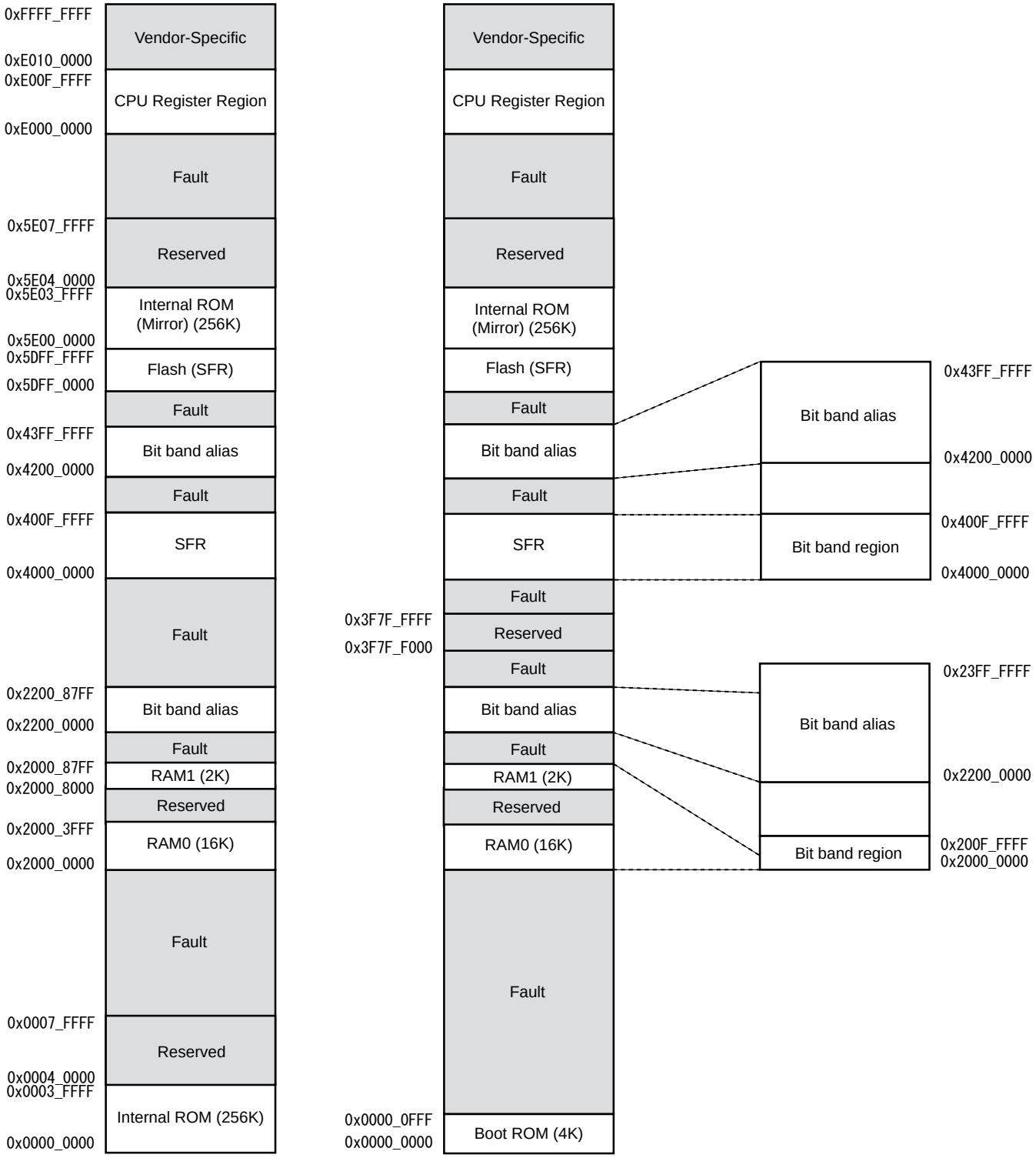
Single chip mode Single boot mode

Figure 4-1 MemoryMap(512KB)



Single chip mode Single boot mode

Figure 4-2 MemoryMap(384KB)



Single chip mode Single boot mode

Figure 4-3 MemoryMap(256KB)

4.2 Bus Matrix

This MCU contains two bus masters such as a CPU core and µDMA controllers .

Bus masters connect to slave ports (S0 to S5) of Bus Matrix. In the bus matrix, master ports (M0 to M15) connect to peripheral functions via connections described as (o) or (・) in the following figure. (・) shows a connection to a mirror area.

While multiple slaves are connected on the same bus master line in the Bus Matrix, if multiple slave accesses are generated at the same time, a priority is given to access from a master with the smallest slave number.

4.2.1 Structure

4.2.1.1 Single chip mode

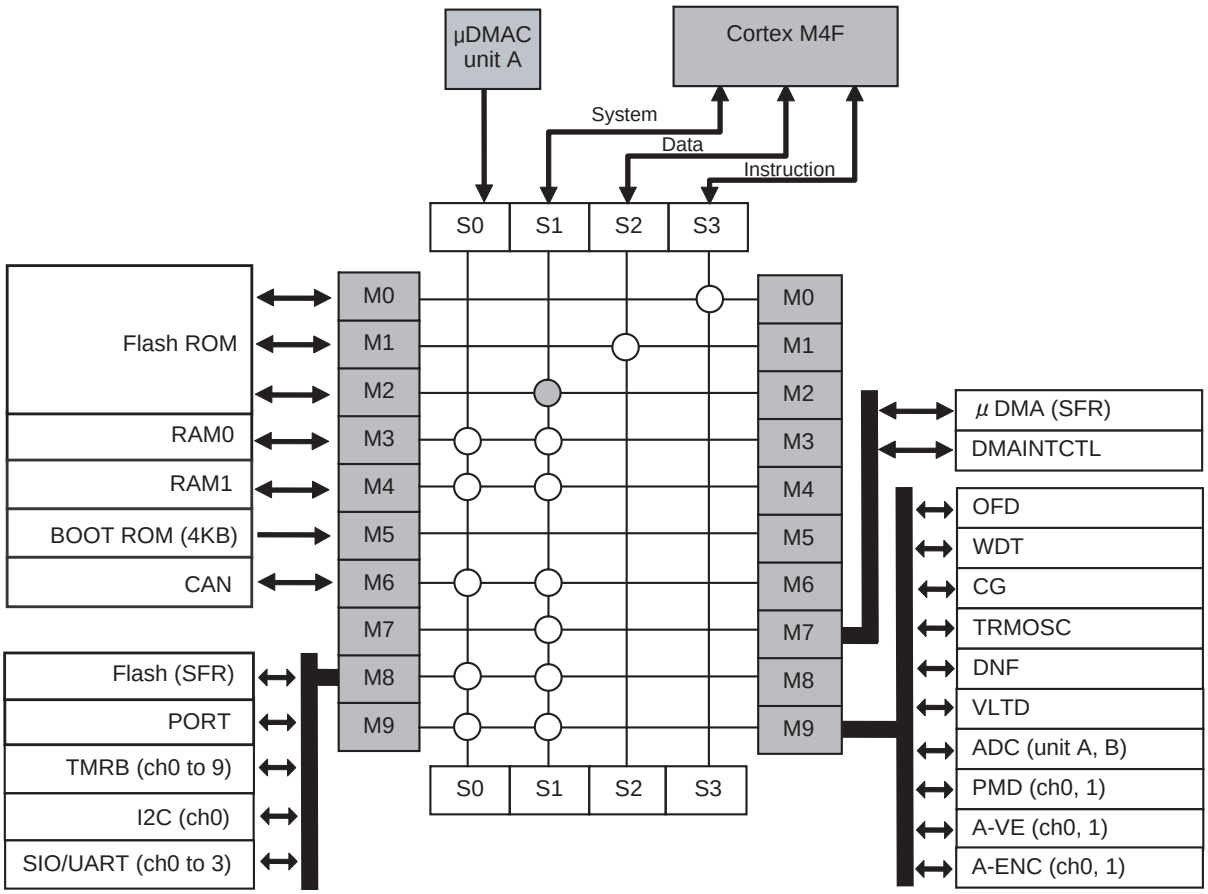


Figure 4-4 Bus Matrix of TMPM475FDFG/FZFG/FYFG

4.2.1.2 Single boot mode

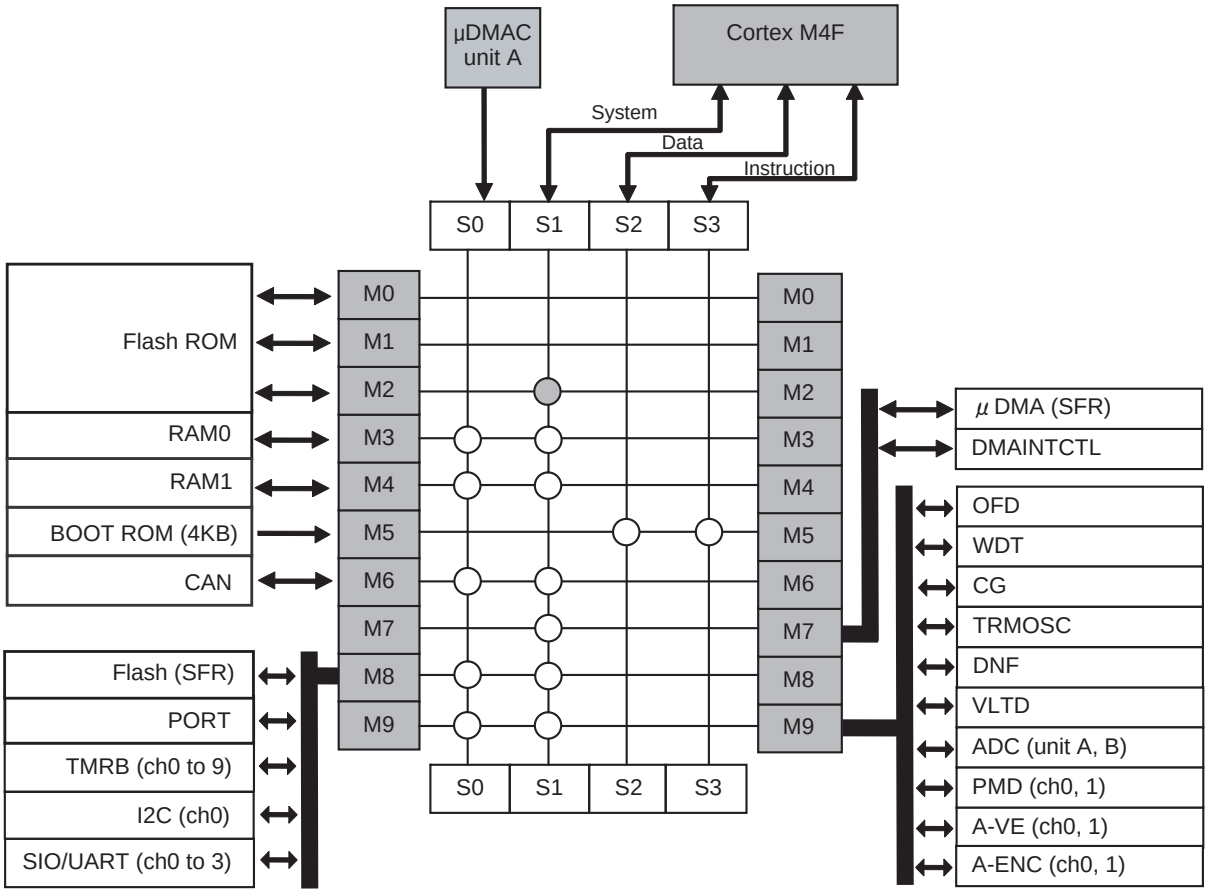


Figure 4-5 Bus Matrix of TMPM475FDFG/FZFG/FYFG

4.2.2 Connection table

4.2.2.1 Code area / SRAM area

(1) Single chip mode

Start Address	Slave	Master	µDMAC unitA	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2	S3
0x0000_0000	Flash ROM	M0 M1	Fault	Fault	o	o
0x0010_0000	Fault	-	Fault	Fault	Fault	Fault
0x2000_0000	RAM0	M3	o	o	Fault	Fault
0x2000_8000	RAM1	M4	o	o	Fault	Fault
0x2000_8800	Fault	-	Fault	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
0x2400_0000	Fault	-	Fault	Fault	Fault	Fault

(2) Single boot mode

Start Address	Slave	Master	µDMAC unitA	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2	S3
0x0000_0000	Boot ROM	M5	Fault	Fault	o	o
0x0000_1000	Fault	-	Fault	Fault	Fault	Fault
0x2000_0000	RAM0	M3	o	o	Fault	Fault
0x2000_8000	RAM1	M4	o	o	Fault	Fault
0x2003_0400	Fault	-	Fault	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
0x2400_0000	Fault	-	Fault	Fault	Fault	Fault
0x3F7F_F000	Reserved	-	Fault	Reserved	Fault	Fault
0x3F80_0000	Fault	-	Fault	Fault	Fault	Fault

Note: Please do not access the address range given in Reserved.

4.2.2.2 Peripheral area / External bus area

Start Address	Slave	Master	µDMAC unitA	Core S-Bus	Core D-Bus	Core I-Bus
			S0	S1	S2	S3
0x4000_0000	Fault	-	Fault	Fault	Fault	Fault
0x4000_5000	CAN	M6	o	o	Fault	Fault
0x4004_C000	µDMAC unitA (SFR)	M7	-	o	Fault	Fault
0x4005_F000	DMAIF	M7	-	o	Fault	Fault
0x400C_0000	PORT	M8	o	o	Fault	Fault
0x400C_4000	TMRB (ch0 to -9)	M8	o	o	Fault	Fault
0x400E_0000	I2C (ch0)	M8	o	o	Fault	Fault
0x400E_1000	SIO/UART(ch0 to 3)	M8	o	o	Fault	Fault
0x400F_1000	OFD	M9	o	o	Fault	Fault
0x400F_2000	WDT	M9	o	o	Fault	Fault
0x400F_3000	CG	M9	o	o	Fault	Fault
0x400F_3200	TRMOSC	M9	o	o	Fault	Fault
0x400F_3400	DNF	M9	o	o	Fault	Fault
0x400F_4000	LVD	M9	o	o	Fault	Fault
0x400F_6000	PMD (ch0,1)	M9	o	o	Fault	Fault
0x400F_7000	A-ENC (ch0,1)	M9	o	o	Fault	Fault
0x400F_8000	A-VE (ch0, 1)	M9	o	o	Fault	Fault
0x400F_D000	ADC (unit A, B)	M9	o	o	Fault	Fault
0x4010_0000	Fault	-	Fault	Fault	Fault	Fault
0x4200_0000	Bit band alias	-	Fault	o	Fault	Fault
0x4400_0000	Fault	-	Fault	Fault	Fault	Fault
0x5DFF_0000	Flash (SFR)	M8	-	o	Fault	Fault
0x5E00_0000	Flash(Mirror)	M2	o	o	Fault	Fault
0x5E08_0000	Fault	-	Fault	Fault	Fault	Fault

4.2.3 Address lists of peripheral functions

Do not access to addresses in the peripheral area except control registers. For details of control registers, refer to Chapter of each peripheral functions.

Peripheral Function		Base Address	Area
CAN Mailbox	ch0	0x4000_5000	SFR0
	ch1	0x4000_5020	SFR0
	ch2	0x4000_5040	SFR0
	ch3	0x4000_5060	SFR0
	ch4	0x4000_5080	SFR0
	ch5	0x4000_50A0	SFR0
	ch6	0x4000_50C0	SFR0
	ch7	0x4000_50E0	SFR0
	ch8	0x4000_5100	SFR0
	ch9	0x4000_5120	SFR0
	ch10	0x4000_5140	SFR0
	ch11	0x4000_5160	SFR0
	ch12	0x4000_5180	SFR0
	ch13	0x4000_51A0	SFR0
	ch14	0x4000_51C0	SFR0
	ch15	0x4000_51E0	SFR0
	ch16	0x4000_5200	SFR0
	ch17	0x4000_5220	SFR0
	ch18	0x4000_5240	SFR0
	ch19	0x4000_5260	SFR0
	ch20	0x4000_5280	SFR0
	ch21	0x4000_52A0	SFR0
	ch22	0x4000_52C0	SFR0
	ch23	0x4000_52E0	SFR0
	ch24	0x4000_5300	SFR0
	ch25	0x4000_5320	SFR0
	ch26	0x4000_5340	SFR0
	ch27	0x4000_5360	SFR0
	ch28	0x4000_5380	SFR0
	ch29	0x4000_53A0	SFR0
	ch30	0x4000_53C0	SFR0
	ch31	0x4000_53E0	SFR0
CAN Controller (CAN)	unit 0	0x4000_5400	SFR0
µDMA Controller (µDMAC)	unit A	0x4004_C000	SFR0
	DMAIF	0x4005_F000	SFR0

Peripheral Function		Base Address	Area
Input/Output Ports	PORTA	0x400C_0000	SFR0
	PORTB	0x400C_0100	SFR0
	PORTC	0x400C_0200	SFR0
	PORTD	0x400C_0300	SFR0
	PORTE	0x400C_0400	SFR0
	PORTF	0x400C_0500	SFR0
	PORTG	0x400C_0600	SFR0
	PORTH	0x400C_0700	SFR0
	PORTJ	0x400C_0800	SFR0
	PORTK	0x400C_0900	SFR0
	PORTL	0x400C_0A00	SFR0
	PORTN	0x400C_0C00	SFR0
	PORTP	0x400C_0D00	SFR0
16-bit Timer/Event Counters (TMRB)	ch0	0x400C_4000	SFR0
	ch1	0x400C_4100	SFR0
	ch2	0x400C_4200	SFR0
	ch3	0x400C_4300	SFR0
	ch4	0x400C_4400	SFR0
	ch5	0x400C_4500	SFR0
	ch6	0x400C_4600	SFR0
	ch7	0x400C_4700	SFR0
	ch8	0x400C_4800	SFR0
	ch9	0x400C_4900	SFR0
Serial Bus Interface (I2C)	ch0	0x400E_0000	SFR0
Serial Channel (SIO/UART)	ch0	0x400E_1000	SFR0
	ch1	0x400E_1100	SFR0
	ch2	0x400E_1200	SFR0
	ch3	0x400E_1300	SFR0
Oscillation frequency detect function (OFD)	unit 0	0x400F_1000	SFR0
Watch Dog Timer (WDT)	ch0	0x400F_2000	SFR0
Clock/Mode control (CG)		0x400F_3000	SFR0
Built-in high-speed oscillation adjustment function (TRMOSC)	unit 0	0x400F_3200	SFR0
Digital Noise Filter Circuit (DNF)	unit 0	0x400F_3400	SFR0
Voltage detect function (VLTD)	unit 0	0x400F_4000	SFR0
Programmable motor driver (PMD)	ch0	0x400F_6000	SFR0
	ch1	0x400F_6100	SFR0
Encoder input circuit (A-ENC)	ch0	0x400F_7000	SFR0
	ch1	0x400F_7100	SFR0
Vector engine (A-VE)	ch0	0x400F_8000	SRF0
	ch1	0x400F_8400	SFR0
Analog/Digital Converter (ADC)	unit A	0x400F_D000	SFR0
	unit B	0x400F_D100	SFR0
Flash / Debug (FC)	ch0	0x5DFF_0000	SFR0

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. TMPM475FDFG/FZFG/FYFG has a function to insert a stable time automatically.Once power voltage is beyond the release voltage of power-on-reset, power counter starts operation, and then after tPWUP (approximately 0.9ms in fosc=10MHz) internal reset signal is released.

TMPM475FDFG/FZFG/FYFG has a function to enable low voltage detection circuit (VLTD) operation. Within tPWUP, if the supply voltage becomes upper than the detection voltage <VDLVL[1:0]>, internal reset signal is released by power counter stops operation.

Power-on-reset circuit operation is referred to Section of " Power-on-reset circuit (POR) ". And low voltage detection circuit operation is referred to Section of " Low voltage detection circuit (VLTD) " .

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

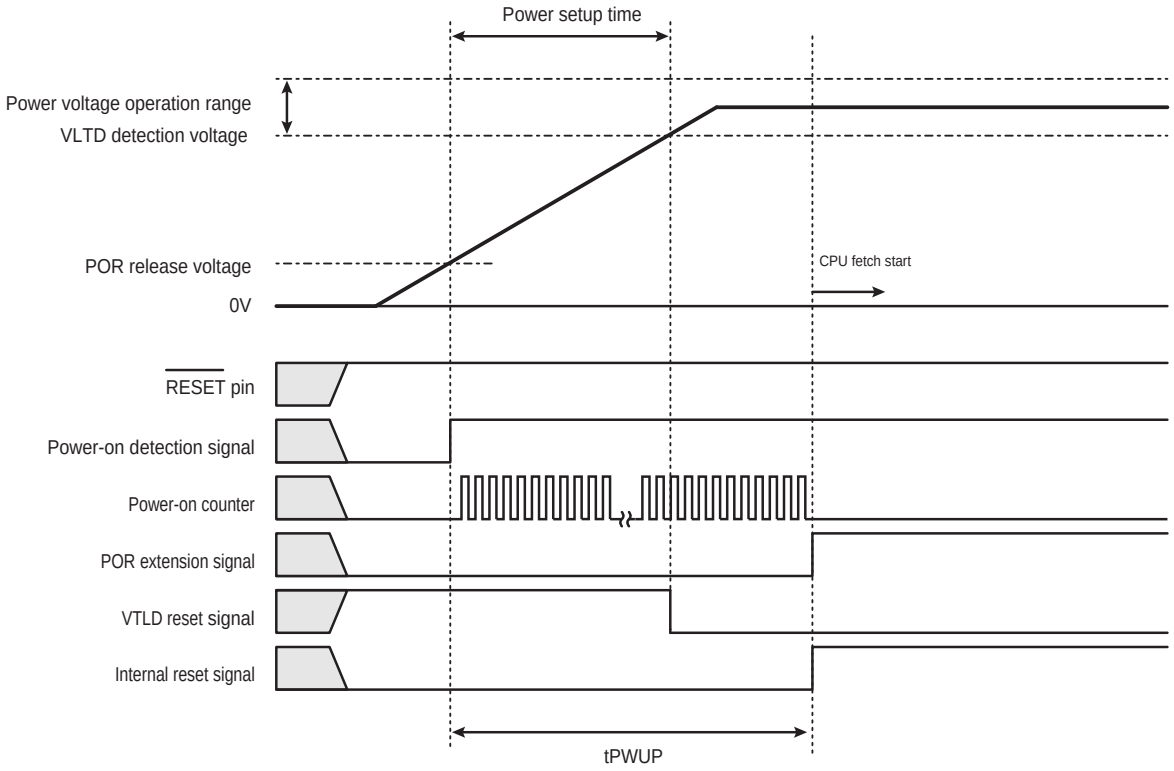


Figure 5-1 Reset Operation by Power-on-reset Circuit (not using $\overline{\text{RESET}}$ pin)

When supply voltage has increased over the VLTD detection voltage while the POR extension signal is at "Low" level, the POR extension signal goes "High" level and this causes the internal reset signal to be "High" level. Subsequently, the reset status is released.

5.1.2 Reset by low-voltage-detection circuit (not using $\overline{\text{RESET}}$ pin)

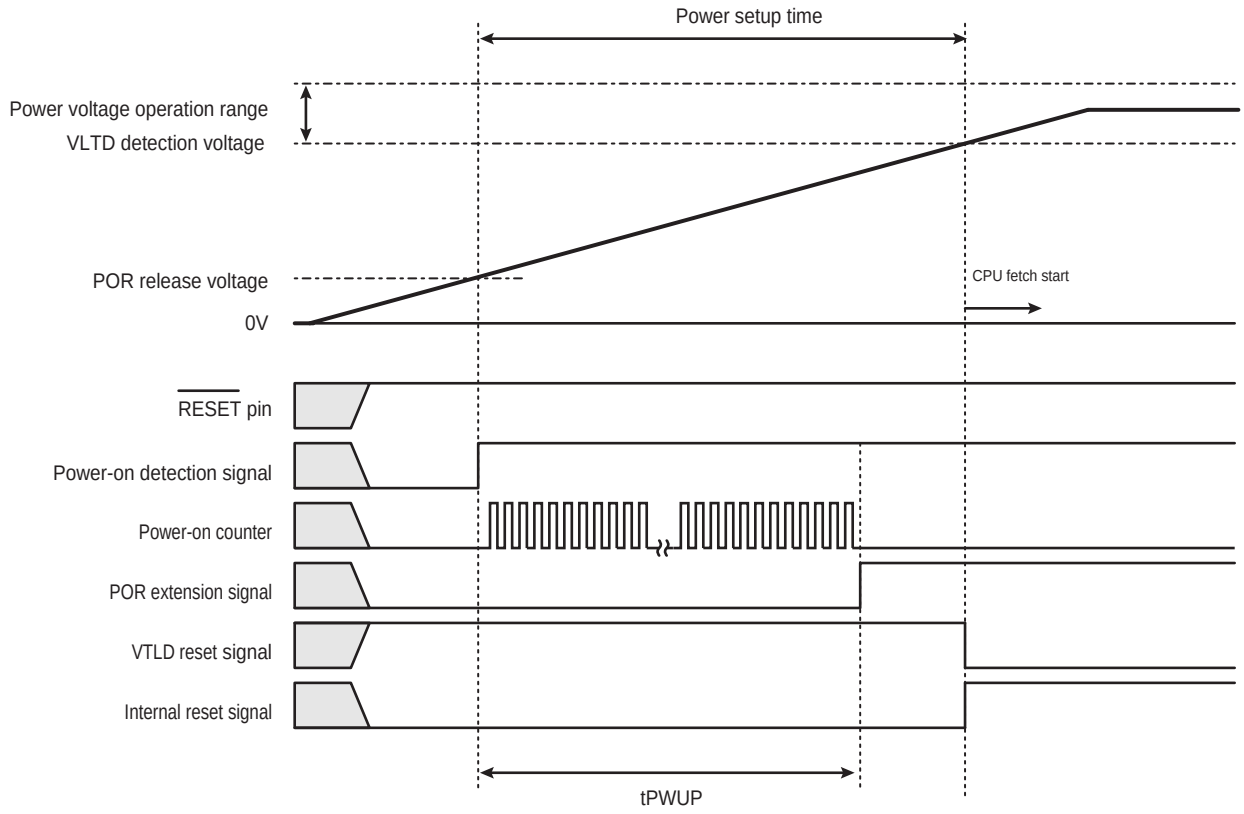


Figure 5-2 Reset Operation by Low-voltage-detection Circuit (not using $\overline{\text{RESET}}$ pin)

When the supply voltage has increased over the VLTD detection voltage while the POR extension signal is at "High" level, when the VLTD reset signal goes "Low" level and this causes the internal reset signal to be "High" level. Subsequently, the reset status is released.

5.1.3 Reset by $\overline{\text{RESET}}$ pin (Reset operation by POR is valid.)

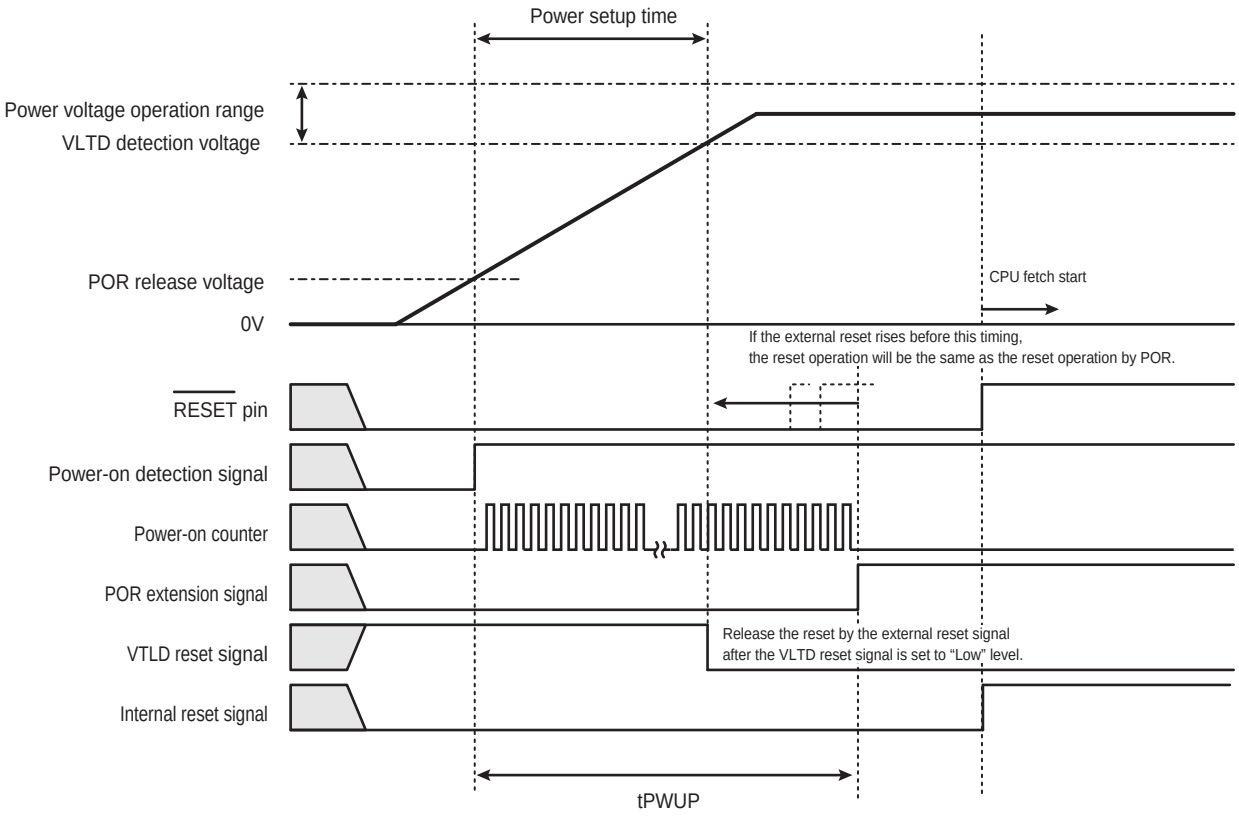


Figure 5-3 Reset Operation by $\overline{\text{RESET}}$ pin (Reset operation by POR is valid.)

While the reset operation by POR is valid, when the RESET pin goes "High" level after the POR extension signal has become "High" level, this causes the internal reset signal to be "High". Subsequently, the reset status is released.

If the RESET signal goes "High" level until immediately before the POR extension signal goes "High" level after the VLTD-reset-signal has become "Low" level, the reset operation will be the same as the operation described in 5.1.1 Reset Operation by POR.

Note that the RESET signal should be "Low" as long as the VLTD reset signal is "High" (i.e. supply voltage is lower than the operation voltage).

5.1.4 Reset by $\overline{\text{RESET}}$ pin (Reset operation by VLTD is valid.)

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 tPWUP after power-on reset signal becomes " High " , the reset process will be the same as the power-on described in 5.1.1.

TMPM475FDFG/FZFG/FYFG has a function to enable low voltage detection circuit (VLTD) operation. Before $\overline{\text{RESET}}$ pin is set to " High " , if the supply voltage becomes upper than the detection voltage <VDLVL [1:0]>, internal reset signal is released after $\overline{\text{RESET}}$ pin is set to " Low " .

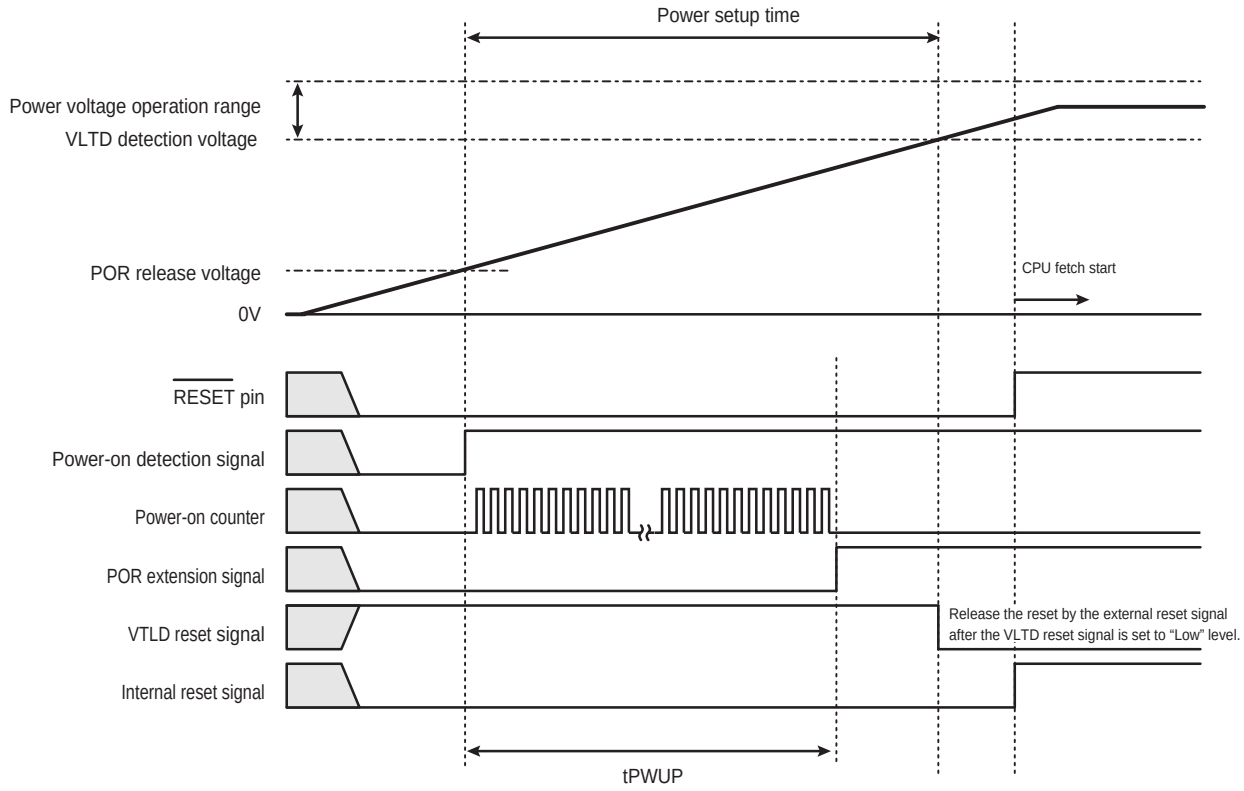


Figure 5-4 Reset Operation by $\overline{\text{RESET}}$ pin (Reset operation by VLTD is valid.)

While the reset operation by VLTD is valid, when the RESET pin and the internal reset signal goes "High" level after the VLTD reset signal has become "Low" level, the reset status is released.

Note that the RESET signal should be "Low" as long as the VLTD reset signal is "High" (i.e. supply voltage is lower than the operation voltage).

5.2 Warm-up

5.2.1 Reset Duration

To do reset TMPM475FDFG/FZFG/FYFG, the following condition is required; power supply voltage is in the operational range; RESET pin is kept " Low " at least for 12 system clocks (minimum duration of 1.2 μsec in fiosc = 10MHz) by internal high frequency oscillator became stable. After $\overline{\text{RESET}}$ pin becomes " High " , internal reset will be released .

5.3 After reset

After reset, the control register of processor core and the peripheral function control register (SFR) are almost 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 circuit controls the internal/external oscillator, clock gear, prescaler clock and warm-up of the PLL clock multiplication circuit and oscillator.

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

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

6.2 Registers

6.2.1 Register List

The tables below show the registers and their addresses related to the clock control.

For the base addresses, refer to the chapter on "A List of the Peripheral Function Base addresses" of "Memory Map."

Peripheral function :CG

Register name		Address (Base+)
System control register	CGSYSCR	0x0000
Oscillation control register	CGOSCCR	0x0004
Standby control register	CGSTBYCR	0x0008
PLL selection register	CGPLLSEL	0x000C
Clock stop register for peripheral	CGCKSTP	0x0040

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 Specifies the source clock to fperiph. Selecting fc fixes fperiph regardless of the clock gear mode.
11	−	R	Read as "0".
10- 8	PRCK[2:0]	R/W	Division ratio selection for the prescaler clock (φT0) 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved Selects a division ratio for the prescaler clock (φT0) supplied to the peripheral functions. Do not change a division ratio for the prescaler clock while the peripheral functions are operating. Note that when a division ratio of the prescaler clock is changed, a frequency of φT0 must be lower than the frequency of the fsys.
7-3	−	R	Read as "0".
2-0	GEAR[2:0]	R/W	Division ratio selection of the clock gear (fgear) 000: fc 100: fc/2 001: Reserved 101: fc/4 010: Reserved 110: fc/8 011: Reserved 111: fc/16 Selects a division ratio of the clock gear. Do not change a division ratio of the clock gear while the peripheral functions are operating. Note that when the clock gear function is used, a frequency of φT0 must be lower than the frequency of the fsys.

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	-	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	Counter compare value setting of the warm-up timer for the high-speed oscillator Sets a compare value for the counter of the warm-up timer. Sets the upper 12 bits values of the 16 bits calculation values of the warm-up time.
19	WUPSEL2	R/W	Selects a source clock of the warm-up timer for the high-speed oscillator (Note 1) 0: Internal (fiosc) 1: External (feosc)
18	−	R/W	Reserved (Even if "0" or "1" is written, there are no troubles in operation.)
17	OSCSEL	R/W	Selection of high-speed oscillator (fosc) 0: Internal (fiosc) 1: External (feosc) Select source clock of high-speed oscillator. When the contents of the register is updated, confirm whether destination clock oscillates stably. Also, if the source clock is changed, check whether the changed value is reflected to <OSCSEL>.
16	XEN2	R/W	Internal high-speed oscillator (IOSC) control 0: Disabled 1: Enabled Controls the internal high-speed oscillator (IOSC). When this bit is enabled, an oscillation stable time is required using the warm-up counter. For details, refer to"6.3.5 Warm-up function".
15-12	−	R/W	Write as "0".
11-10	−	R	Read as "0".
9	−	R/W	Write as "0".
8	XEN1	R/W	External high-speed oscillator (EOSC) control 0: Disabled 1: Enabled Controls the external high-speed oscillator. When this bit is enabled, an oscillation stable time is required using the warm-up counter. For details, refer to"6.3.5 Warm-up function".
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 control 0: Stop 1: Oscillation Controls the PLL operation. When this bit is enabled, an oscillation stable time is required using the warm-up counter. For details, refer to"6.3.5 Warm-up function".

Bit	Bit Symbol	Type	Function
1	WUEF	R	Status of warm-up timer for high-oscillator 0: Warm-up completed. 1: Warm-up operation Enable to monitor the status of the warm-up timer.
0	WUEON	W	The warm-up timer control for the high-speed oscillator 1: The operation starts. Check whether <WUEF> is "0" before the warm-up timer operation starts (set "1"). Writing "0" to <WUON> has no meaning and "0" is read.

Note: Before changing to STOP mode, set up the same clock selected by <OSCSEL>.

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	-	-	-	-	-	-	-	-
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".
16-10	−	R	Read as "0".
9	−	R/W	Write as "0".
8	RXEN	R/W	High-speed oscillator operation after releasing the STOP mode. Write as "1".
7-3	−	R	Read as "0".
2-0	STBY[2:0]	R/W	Low power consumption mode 000: Reserved 001: STOP 010: Reserved 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved To enter the STOP mode, disable the oscillation (IOSC or EOSC) 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	PLLSET							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PLLSET							PLLSEL
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-1	PLLSET	R/W	PLL multiplying value (Do not use except below) 0x591E: Input clock 10MHz, output clock 80MHz 0x5926: Input clock 10MHz, output clock 100MHz 0x59AE: Input clock 10MHz, output clock 120MHz
0	PLLSEL	R/W	Selects a source clock of the high-speed clock (fc) 0: fosc use 1: f_{PLL} Use This bit is used to select the source clock for the high-speed clock (fc).

6.2.6 CGCKSTP (Clock stop register for peripheral)

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

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2	CANSTP	R/W	CAN clock control 0 : operation 1 : clock stop Enable / disable CAN clock
1-0	−	R	Read as "0".

Note: Set CGCKLSTP<CANSTP> to "1" from "0" after confirming stop of a circuit operation.

6.3 Clock control

6.3.1 Clock Type

Each clock is defined as follows :

feosc	: Clock input from external high-speed oscillator (EOSC)
fiosc	: Clock input from internal high-speed oscillator (IOSC)
fosc	: Clock input from high-speed oscillator
f_{PLL}	: Clock input from PLL circuit (x8, x10. x12)
fc	: Clock selected by fosc or f_{PLL} (high-speed clock)
fgear	: Clock divided from the high-speed clock
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.

High-speed oscillator (EOSC)	: Stop
High-speed oscillator (IOSC)	: Oscillating
Phase locked loop circuit (PLL)	: Stop
High-speed clock gear	: fc (no frequency dividing)

Reset operation causes all the clock configurations to be the same as fosc.

$f_C = f_{OSC}$

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

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

φT0 = f_{periph} (= f_{OSC})

6.3.3 Clock system Diagram

Figure 6-1 shows the clock system diagram.

After reset, the clock indicated by the arrow is selected among input clocks supplied to the selector.

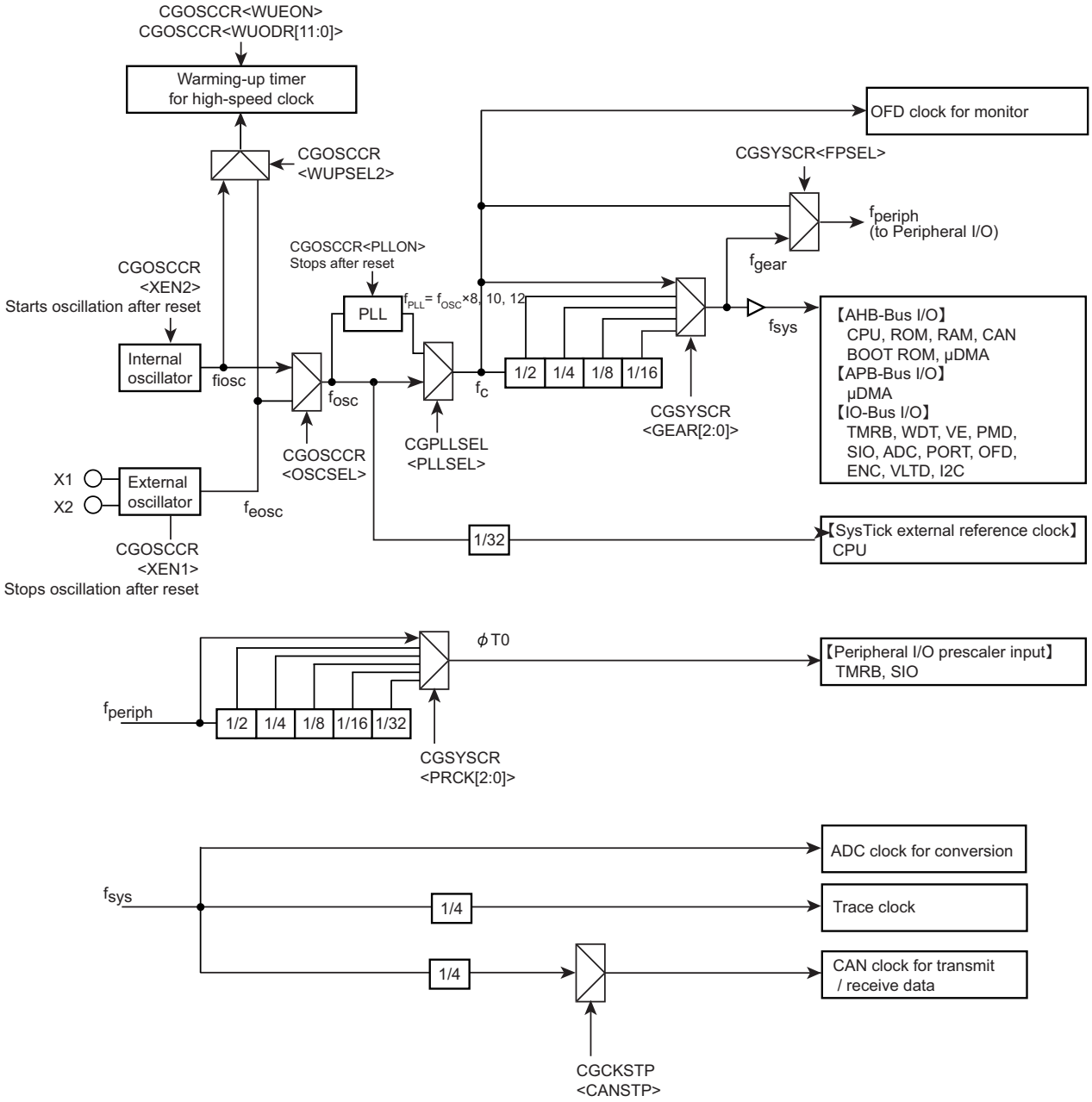


Figure 6-1 Clock Block Diagram

6.3.4 Clock Multiplication Circuit (PLL)

This circuit outputs the f_{PLL} clock that is multiplication (x8, x10, x16) 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 multiplication (x8, x10, x16) 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

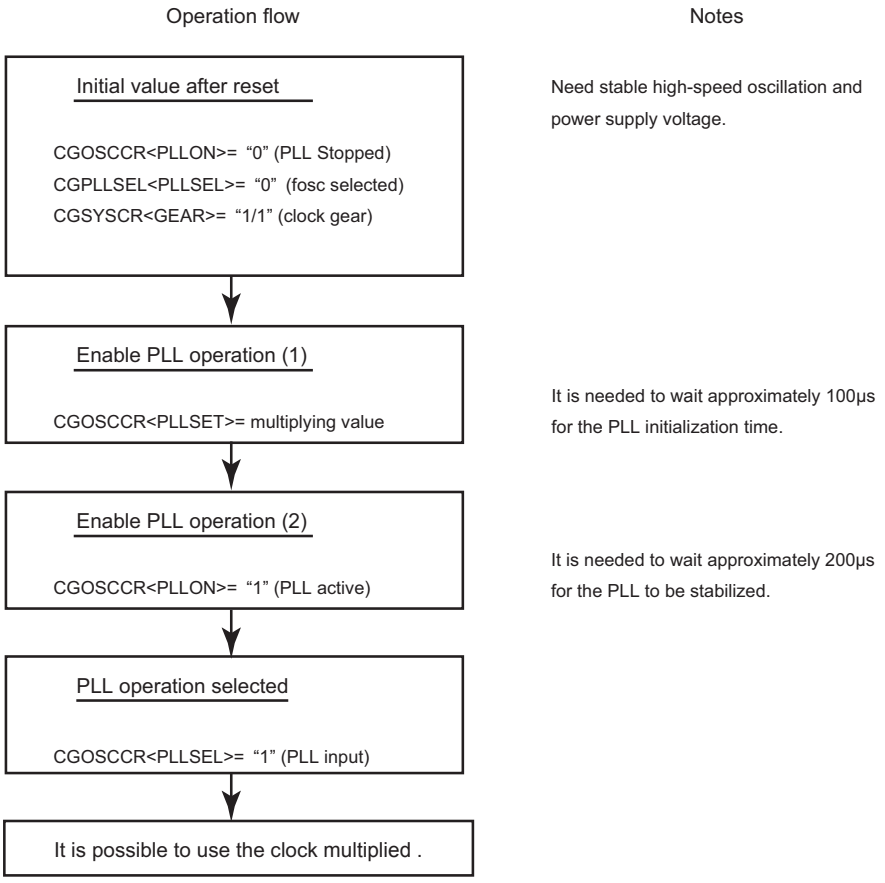
The PLL requires a certain amount of time to be stabilized, which should be secured using the warmup function or other methods.

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.



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 the low power consumption mode triggers the automatic timer count. After the specified time is reached, the system clock is output and the CPU starts operation.

How to configure the warm-up function.

1. Specify the count up clock

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

2. Specify the warm-up counter value

   The warm-up time can be selected by setting the CGOSCCR<WUODR[11:0]>. Therefore, the lower 4 bits of the calculated value of the following equation are rounded down. The upper 12 bits are set to CGxWUHCR<WUPT[11:0]>.

   The following shows the warm-up setting and example.

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

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

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

Drop the last 4 bits, set 0x9C4 into the CGOSCCR<WUPT[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 = feosc)

CGOSCCR<WUPSEL1> = "0"	: Write "0" to CGOSCCR<WUPSEL1>
CGOSCCR<WUPSEL2> = "1"	: Specify the clock source for warm-up timer (1: External (feosc))
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 TMPM475FDFG/FZFG/FYFG 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 : 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. feosc fiosc	multi-plica-tion	Min oper-ating freq.	Max oper-ating 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
10	8	1.25	80	10	80	40	20	10	5	10	5	2.5	1.25	-
10	10	1.25	100	10	100	50	25	12.5	6.25	10	5	2.5	1.25	-
10	12	1.25	120	10	120	60	30	15	7.5	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 is OFF.

Note 4: ."-" : Reserved

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

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

1. CGOSCCR<WUODR[11:0]> = "Warm-up time" : Set Warm-up time.
2. CGOSCCR<XEN1> = "1" : Enable the external oscillator.
3. CGOSCCR<WUPSEL2> = "1" : Specify the external oscillator clock as source clock for warm-up counter.
4. CGOSCCR<WUEON>="1" : Enable warm-up counting (WUP)

   Read CGOSCCR<WUEF> : Wait until the state becomes "0" (warm-up is finished)
5. CGOSCCR<OSCSEL> = "1" : Switch the system clock to the external oscillator.
6. Read CGOSCCR<OSCSEL> : Confirm CGOSCCR[17]<OSCSEL> become "1". (External oscillator is selected.)
7. CGOSCCR<XEN2> = "0" : Internal oscillator is disabled.

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

6.4 Modes and Mode Transitions

6.4.1 Mode Transitions

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

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

Figure 6-2 shows mode transition diagram.

For a detail of WFI instruction or Sleep-on-exit, refer to the documentation set issued by ARM Limited.

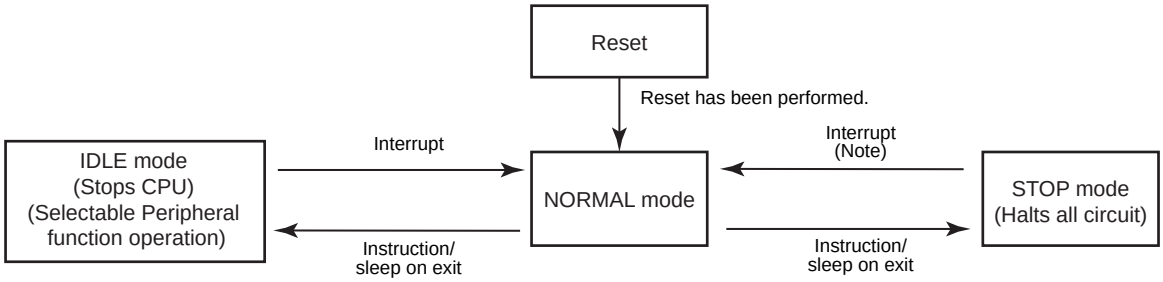


Figure 6-2 Mode Transition Diagram

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

6.5.1 NORMAL mode

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

It is shifted to the NORMAL mode after reset.

6.6 Low Power Consumption Modes

The TMPM475FDFG/FZFG/FYFG has two low power consumption modes: IDLE and STOP. To shift to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY[2:0]> and execute the WFI (Wait For Interrupt) instruction.In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance. See the chapter "Exceptions" for details.

Note 1: The TMPM475FDFG/FZFG/FYFG does not offer any event for releasing the low power consumption mode. Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited.

Note 2: The TMPM475FDFG/FZFG/FYFG does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the <SLEEPDEEP> bit of the system control register is prohibited.

Note 3: Do not shift to the low power consumption mode, during operating warm-up timer.

The features of each mode are described as follows.

6.6.1 IDLE Mode

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

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

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

6.6.2 STOP mode

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

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

Table 6-2 shows the pin status in the STOP mode.

Table 6-2 Pin States in the STOP mode

Function	Pin name	I/O	STOP (Note)
Control	$\overline{\text{RESET}}$, MODE	Input	Enable
Oscillator	X1	Input	Disable
	X2	Output	"High" level output
Port	PAx to PPx	Input	Depend on PxIE[m]
		Output	Depend on PxCR[m]
Debug	SWCLK, SWDIO $\overline{\text{TRST}}$, TCK, TMS, TDI	Input	Depend on PxIE[m]
	SWDIO, SWV TDO, TRACECLK TRACEDATA0/1/2/3	Output	Depend on PxCR[m] and enable when data is valid
Interrupt	INT0 to INTF	Input	Depend on PxIE[m]
PMD	UO0, VO0, WO0, XO0, YO0, ZO0 UO1, VO1, WO1, XO1, YO1, ZO1	Output	Depend on PxCR[m]
Except above	Except above.	Input	Depend on PxIE[m]
	Except above.	Output	Depend on PxCR[m]

(Note) x: port number / m: corresponding bit

6.6.3 Low power Consumption Mode Setting

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

Table 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.

Table 6-4 Operational Status in Each Mode

Block	NORMAL	IDLE	STOP
Processor core	o	-	-
µDMAC	o	o	-
I/O port	o	o	- (Note1)
PMD	o	o	-
A-ENC	o	o	-
OFD	o	o	Δ
ADC	o	o	-
CAN	o	o	-
A-VE	o	•	-
SIO/UART	o	•	-
I2C	o	•	-
TMRB	o	•	-
WDT	o	Δ	Δ
VLTD	o	o	o (Note2)
POR	o	o	o (Note2)
DNF	o	o	-
CG	o	o	-
PLL	o	o	Δ
High-speed oscillator (fc)	o	o	-

o : Operation is available when in the target mode.

• : Operation is available when in the target mode(The operation / stop can be selected for each channel)

- : The clock to module stops automatically when transiting to the target mode.

Δ: Enables to select enabling or disabling module operation by software when in the target mode.

Note 1: It depends on the port register.

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

6.6.5 Releasing the Low Power Consumption Mode

The low power consumption mode can be released by an interrupt request or reset. The release source that can be used is determined by the low power consumption mode selected. Details are shown in Table 6-5.

Table 6-5 Release Source in Each Mode

Low power consumption mode			IDLE	STOP
Release source	Interrupt	INT0 to INTF (Note1)	o	o
		INTENC00, INTENC01 INTENC10, INTENC11	o	×
		INTVCN0, INTVCT0 INTVCN1, INTVCT1	o	×
		INTPWM0, INTEMG0, INTOVV0 INTPWM1, INTEMG1, INTOVV1	o	×
		INTRX0 to 3, INTTX0 to 3	o	×
		INTADACPA, INTADBCPA INTADACPB, INTADBCPB INTADATMR, INTADBTMR INTADASFT, INTADBSFT INTADAPDA, INTADBPDA INTADAPDB, INTADBPDB	o	×
		INTTB00 to INTTB90 INTCAP00 to INTCAP90 INTTB01 to INTTB91 INTTCP01 to INTCAP91	o	×
		INTDMACATC, INTDMACAERR	o	×
		INTCANRX, INTCANTX, INTCANGB	o	×
	SysTick		o	×
	RESET (OFD)		o	×
	RESET ($\overline{\text{RESET}}$ pin, POR, VLTD)		o	o

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

× : Unavailable

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

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

Note 3: Refer to "6.6.6 Warm-up" about warm-up time.

- Release by interrupt request

  To release the low power consumption mode by an interrupt, the CPU must be set in advance to detect the interrupt. In addition to the setting in the CPU, the clock generator must be set to detect the interrupt to be used to release the STOP modes. And the digital noise filter circuit should be set to disable as well.

- Release by reset

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

  IDLE mode can be released by reset from OFD.

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

- Release by SysTick interrupt

  SysTick interrupt can only be used in IDLE mode.

Refer to "Interrupts" for detail.

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> (Note2) before executing the instruction to enter the STOP mode.

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

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

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

Table 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

Note: When releasing by reset is executed, automatic warm-up is not performed. Input a reset until the oscillator becomes stable.

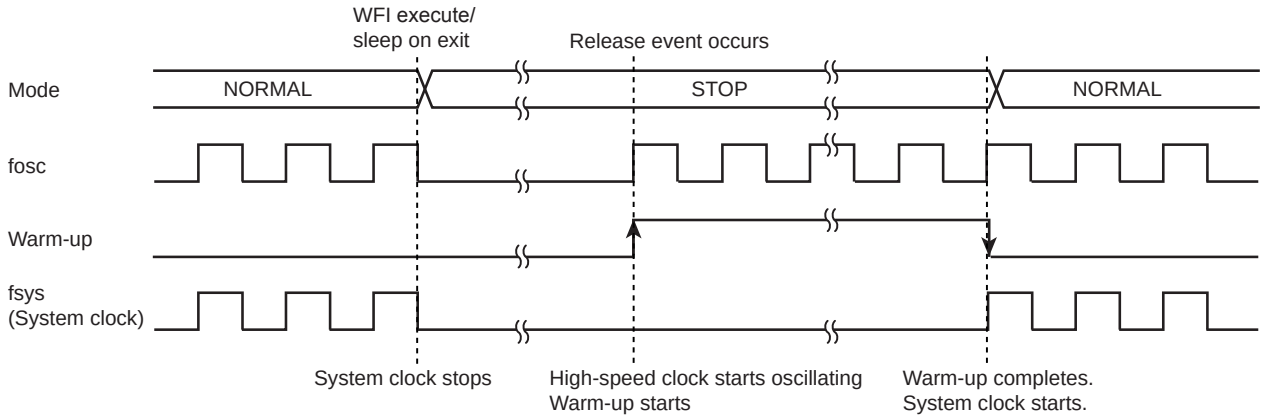
6.6.7 Clock Operation in Mode Transition

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

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



7. Internal High-speed Oscillation Adjustment Function

TMPM475FDFG/FZFG/FYFG has the internal high-speed oscillation adjustment function.

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.

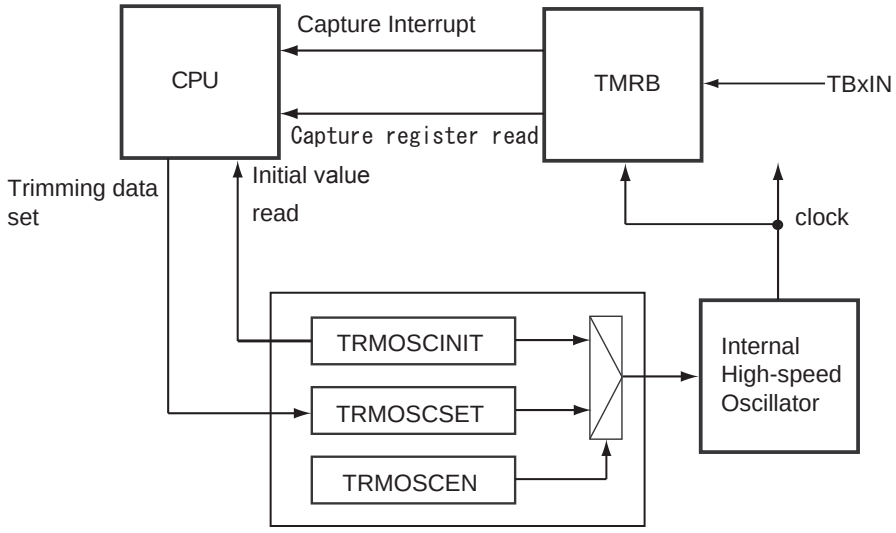


Figure 7-1 Function block diagram

7.2 Registers

7.2.1 Register list

The 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+)
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, TRMOSCxEN, TRMOSCxINIT and TRMOSCxSET 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 TRIMOSCxINIT to a value of TRMOSCxSET.

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	1	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[2:0]	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, approximately −19% to +32% adjustment by the average step of 0.8% is feasible.

In the fine trimming, −0.8% to +0.7% adjustment by 0.1%-step is feasible.

Table 7-1 shows an adjustment range.

Note 1: Each step value is assumed based on the typical condition. In the coarse trimming, it has ±0.6% margin of error. In the fine trimming, it has ±0.1% margin of error.

Note 2: The resolution of each trimming has some variations according to sample or condition.

Table 7-1 Adjustment range

Coarse trimming	
<TRIMSETC[5:0]>	Frequency change (typ.)
011111	+32.0%
011110	+30.6%
011101	+29.2%
011100	+27.8%
011011	+26.5%
・	・
000011	+2.3%
000010	+1.5%
000001	+0.8%
000000	±0%
111111	−0.7%
111110	−1.5%
・	・
100100	−16.9%
100011	−17.4%
100010	−17.9%
100001	−18.3%
100000	−18.8%

Fine trimming	
<TRIMSETF[3:0]>	Frequency change (typ.)
0111	+0.7%
・	・
0001	+0.1%
0000	±0%
1111	−0.1%
1110	−0.2%
・	・
1000	−0.8%

8. Exceptions

This chapter describes features, types and handling of exceptions.

Exceptions have close relation to the CPU core. Refer to "ARM documentation set for the ARM Cortex-M4F" if needed.

8.1 Overview

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

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

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

8.1.1 Exception Types

The following types of exceptions exist in the Cortex-M4F.

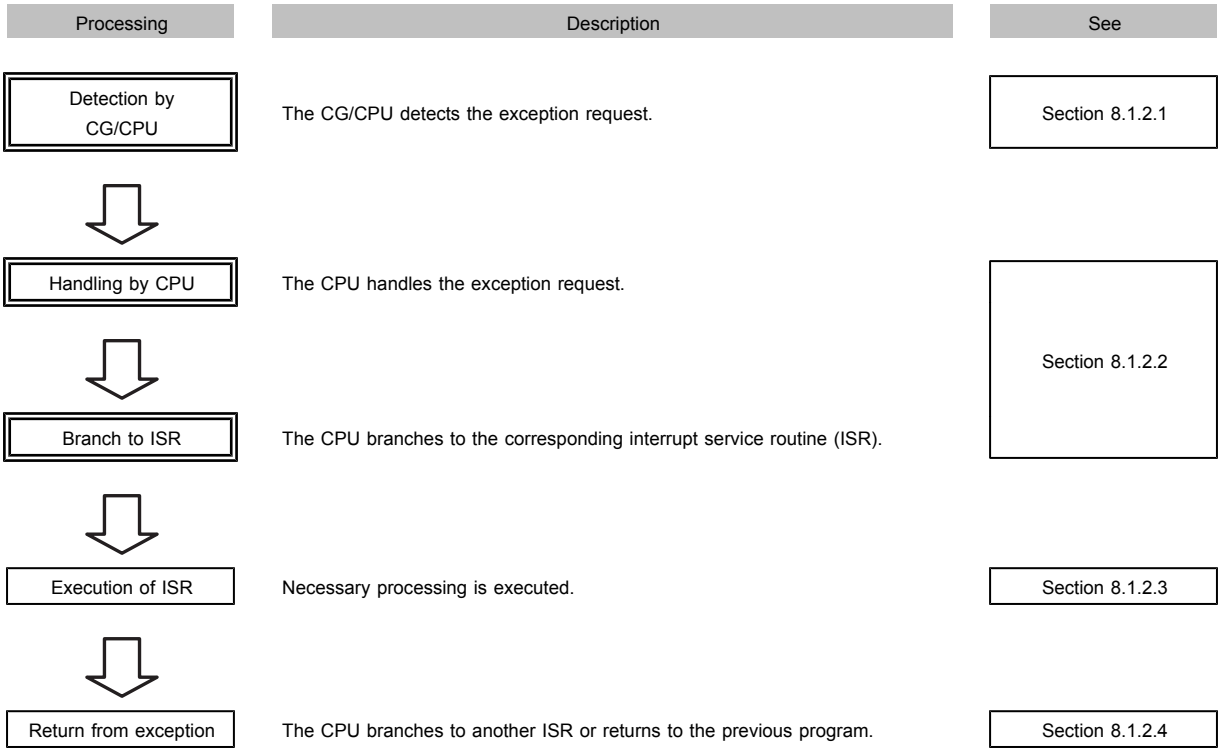
For detailed descriptions on each exception, refer to "ARM documentation set for the ARM Cortex-M4F".

- 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.



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)	$\overline{\text{Reset}}$ 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	Pending system service request
15	SysTick	Configurable	Notification from system timer
up to 16	External Interrupt	Configurable	External interrupt pin or peripheral function (Note 2)

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

Note 2: **External interrupts have different sources and numbers in each product. For details, see "8.5.1.5 List of Interrupt Sources".**

(3) Priority setting

- Priority levels

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

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

TMPM475FDFG/FZFG/FYFG has 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 the number, the higher the priority becomes.

- Priority grouping

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

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

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

Table 8-2 Priority grouping setting

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

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

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

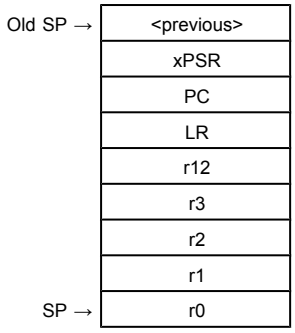
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine (ISP). This is called "pre-emption".

(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 by the completion of the stack push.The following shows the state of the stack after the register contents have been pushed.



(2) Fetching an ISR

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

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

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

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

Set ISR addresses for other exceptions if necessary.

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

8.1.2.3 Executing an ISR

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

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

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

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

8.1.2.4 Exception exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

(2) Exception exit sequence

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

- Pop eight registers

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

- Load current active interrupt number

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

- Select SP

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

8.2 Reset Exceptions

Reset exceptions are generated from the following sources.

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

- External $\overline{\text{RESET}}$ pin

  A reset exception occurs when an external $\overline{\text{RESET}}$ pin changes from "Low" to "High".

- Reset exception by POR

  The power on reset (POR) has a reset generating feature. For details. see the chapter on the POR.

- Reset exception by VLTD

  The low voltage detection circuit (VLTD) has a reset generating feature. For details, see the chapter on the VLTD.

- Reset exception by OFD

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

- 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.

8.4 SysTick

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

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

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

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

8.5 Interrupts

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

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

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

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

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 (route 1).

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

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

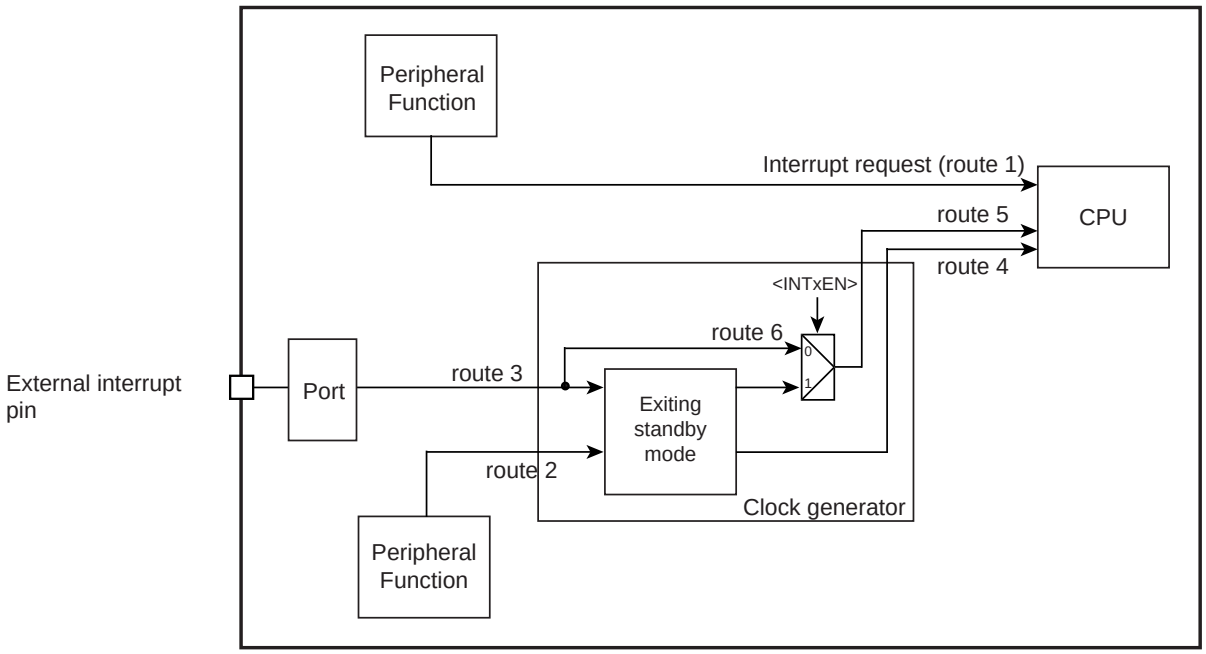


Figure 8-1 Interrupt Sources Route

8.5.1.2 Generation Interrupt Request

Interrupt request are generated by the setting of the external interrupt pins allocated to interrupt factors, peripheral functions, or Interrupt Set-Pending Register.

- Interrupt by the external interrupt pins
- Interrupts by the peripheral functions

  When the interrupts by the peripheral functions are used, interrupt generation must be enabled in the peripheral function to be used.

  For details of the setting, refer to each chapter of the peripheral functions.
- Forcible generation of the interrupts

  To forcibly generate interrupt requests, set the corresponding bit of the interrupt hold setting register in the NVIC.

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

8.5.1.3 Transmission

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

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

8.5.1.4 Precautions when using external interrupt pins

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

If input disabled (PxIE<PxmIE>="0"), inputs from external interrupt pins are "High". Also, if external interrupts are not used as a trigger to release low-power consumption (route 5 of "Figure 8-1 Interrupt Route"), input signals from the external interrupt pins are directly sent to the CPU. Since the CPU recognizes "High" input as an interrupt, interrupts occur if corresponding interrupts are enabled by the CPU as inputs are being disabled.

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

8.5.1.5 List of Interrupt Sources

Table 8-3 shows the list of interrupt sources.

Table 8-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing low-power consumption)	CG interrupt mode control register
0	INT0	External interrupt pin 0	High/Low Edge/Level Selectable	CGIMCGA
1	INT1	External interrupt pin 1		
2	INT2	External interrupt pin 2		
3	INT3	External interrupt pin 3		
4	INT4	External interrupt pin 4	High/Low Edge/Level Selectable	CGIMCGB
5	INT5	External interrupt pin 5		
6	INTRX0	Serial receive interrupt (Channel 0)		

Table 8-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing low-power consumption)	CG interrupt mode control register
7	INTTX0	Serial transmit interrupt (Channel 0)		
8	INTRX1	Serial receive interrupt (Channel 1)		
9	INTTX1	Serial transmit interrupt (Channel 1)		
10	INTVCN0	Vector engine interrupt 0		
11	INTVCN1	Vector engine interrupt 1		
12	INTEMG0	PMD0 EMG interrupt		
13	INTEMG1	PMD1 EMG interrupt		
14	INTOVV0	PMD0 OVV interrupt		
15	INTOVV1	PMD1 OVV interrupt		
16	INTADAPDA	ADCA conversion triggered by PMD is finished The interrupt when choosing INTADxPDA in ADCA		
17	INTADBPDA	ADCB conversion triggered by PMD is finished The interrupt when choosing INTADxPDA in ADCB		
18	INTADAPDB	ADCA conversion triggered by PMD is finished The interrupt when choosing INTADxPDB in ADCA		
19	INTADBPDB	ADCB conversion triggered by PMD is finished The interrupt when choosing INTADxPDB in ADCB		
20	INTTB00	16bit TMRB0 compare match detection 0/ Over flow		
21	INTTB01	16bit TMRB0 compare match detection 1		
22	INTTB10	16bit TMRB1 compare match detection 0/ Over flow		
23	INTTB11	16bit TMRB1 compare match detection 1		
24	INTTB40	16bit TMRB4 compare match detection 0/ Over flow		
25	INTTB41	16bit TMRB4 compare match detection 1		
26	INTTB50	16bit TMRB5 compare match detection 0/ Over flow		
27	INTTB51	16bit TMRB5 compare match detection 1		
28	INTPMD0	PMD0 PWM interrupt		
29	INTPMD1	PMD1 PWM interrupt		
30	INTCAP00	16bit TMRB0 input capture 0		
31	INTCAP01	16bit TMRB0 input capture 1		
32	INTCAP10	16bit TMRB1 input capture 0		
33	INTCAP11	16bit TMRB1 input capture 1		
34	INTCAP40	16bit TMRB4 input capture 0		
35	INTCAP41	16bit TMRB4 input capture 1		
36	INTCAP50	16bit TMRB5 input capture 0		
37	INTCAP51	16bit TMRB5 input capture 1		
38	INT6	External interrupt pin 6	High/Low Edge/Level Selectable	CGIMCGB
39	INT7	External interrupt pin 7		
40	INTRX2	Serial receive interrupt (Channel 2)		
41	INTTX2	Serial transmit interrupt (Channel 2)		
42	INTADACPA	ADCA conversion monitoring function interrupt A		
43	INTADBCPA	ADCB conversion monitoring function interrupt A		
44	INTADACPB	ADCA conversion monitoring function interrupt B		
45	INTADBCPB	ADCB conversion monitoring function interrupt B		
46	INTTB20	16bit TMRB2 compare match detection 0/ Over flow		
47	INTTB21	16bit TMRB2 compare match detection 1		
48	INTTB30	16bit TMRB3 compare match detection 0/ Over flow		

Table 8-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing low-power consumption)	CG interrupt mode control register
49	INTTB31	16bit TMRB3 compare match detection 1		
50	INTCAP20	16bit TMRB2 input capture 0		
51	INTCAP21	16bit TMRB2 input capture 1		
52	INTCAP30	16bit TMRB3 input capture 0		
53	INTCAP31	16bit TMRB3 input capture 1		
54	INTADASFT	ADCA conversion started by software is finished		
55	INTADBSFT	ADCB conversion started by software is finished		
56	INTADATMR	ADCA conversion triggered timer is finished		
57	INTADBTMR	ADCB conversion triggered timer is finished		
58	INT8	External interrupt pin 8	High/Low Edge/Level Selectable	CGIMCGC
59	INT9	External interrupt pin 9		
60	INTA	External interrupt pin A		
61	INTB	External interrupt pin B		
62	INTENC00	Encoder input 0 interrupt 0		
63	INTENC01	Encoder input 0 interrupt 1		
64	INTRX3	Serial receive interrupt (Channel 3)		
65	INTTX3	Serial transmit interrupt (Channel 3)		
66	INTTB60	16bit TMRB6 compare match detection 0/ Over flow		
67	INTTB61	16bit TMRB6 compare match detection 1		
68	INTTB70	16bit TMRB7 compare match detection 0/ Over flow		
69	INTTB71	16bit TMRB7 compare match detection 1		
70	INTCAP60	16bit TMRB6 input capture 0		
71	INTCAP61	16bit TMRB6 input capture 1		
72	INTCAP70	16bit TMRB7 input capture 0		
73	INTCAP71	16bit TMRB7 input capture 1		
74	INTC	External interrupt pin C	High/Low Edge/Level Selectable	CGIMCGD
75	INTD	External interrupt pin D		
76	INTE	External interrupt pin E		
77	INTF	External interrupt pin F		
78	INTVCT0	Vector Engine task termination interrupt 0		
79	INTVCT1	Vector Engine task termination interrupt 1		
80	INTSBI0	I2C interrupt 0		
81	INTCANRX	CAN reception		
82	INTCANTX	CAN transmission		
83	INTCANGB	CAN status		
84	INTTB80	16bit TMRB8 compare match detection 0/ Over flow		
85	INTTB81	16bit TMRB8 compare match detection 1		
86	INTTB90	16bit TMRB9 compare match detection 0/ Over flow		
87	INTTB91	16bit TMRB9 compare match detection 1		
88	INTCAP80	16bit TMRB8 input capture 0		
89	INTCAP81	16bit TMRB8 input capture 1		
90	INTCAP90	16bit TMRB9 input capture 0		
91	INTCAP91	16bit TMRB9 input capture 1		
92	INTDMACATC	DMA transmit interrupt		
93	INTDMACAERR	DMA error interrupt		
94	INTENC10	Encoder input 1 interrupt 0		

Table 8-3 List of Interrupt Sources

No.	Interrupt Source		active level (Clearing low-power consumption)	CG interrupt mode control register
95	INTENC11	Encoder input 1 interrupt 1		

8.5.1.6 Active level

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

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

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

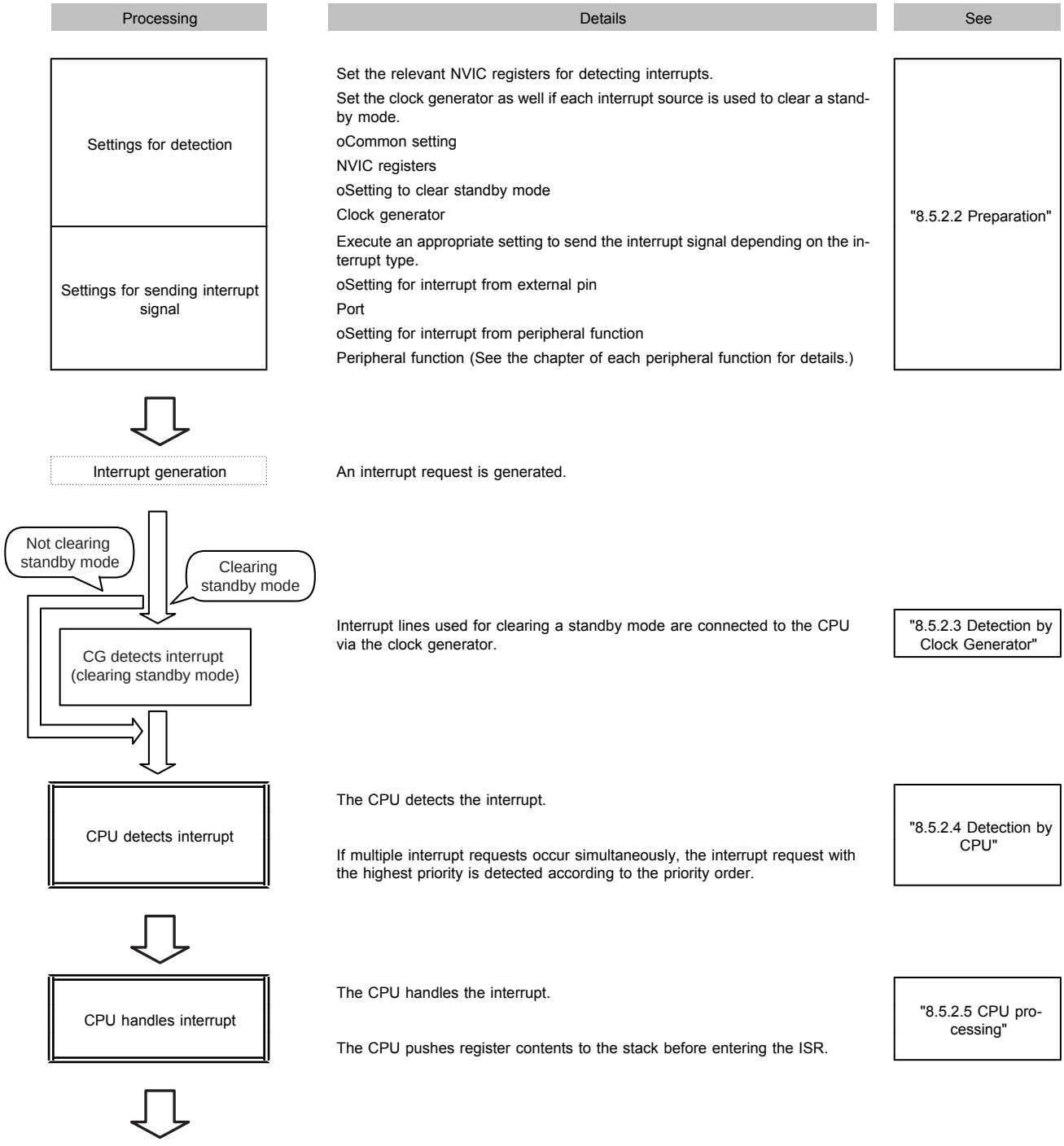
An interrupt request detected by the clock generator is notified to the CPU with a signal in "High" level.

8.5.2 Interrupt Handling

8.5.2.1 Flowchart

The following shows how an interrupt is handled.

In the following descriptions, ▭ indicates hardware handling. ▭ indicates software handling.



Processing	Details	See
ISR execution	Program for the ISR. Clear the interrupt source if needed.	"8.5.2.6 Interrupt Service Routine (ISR)"
Return to preceding program	Configure to return to the preceding program of the ISR.	

8.5.2.2 Preparation

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

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

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

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

1. Disabling interrupt by CPU
2. CPU registers setting
3. 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. Enabling interrupt by CPU

(1) Disabling interrupt by CPU

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

Use "MSR" instruction to set this register.

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

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

(2) CPU registers setting

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

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

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

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

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

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

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

Port register		
PxIE<PxmIE>	←	"1"

Note: **x: port number / m: corresponding bit**
Setting PxIE to enable input enables the corresponding interrupt input. Be careful not to enable interrupts that are not used.

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

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

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

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

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

Note: **m: corresponding bit**

(6) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt.To clear corresponding interrupt request, write a value corresponding to the interrupt to be used to the CGICRCG register.See "8.6.3.2 CGICRCG(CG Interrupt Request Clear Register)" for each value.

Interrupt requests from external pins can be used without setting the clock generator if they are not used for exiting a low-power consumption mode. However, an "High" pulse or "High"-level signal must be input so that the CPU can detect it as an interrupt request. Also, be aware of the description of "8.5.1.4Precautions when using external interrupt pins".

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

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

(7) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

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

At the end, PRIMASK register is zero cleared.

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

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

8.5.2.3 Detection by Clock Generator

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

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

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

8.5.2.4 Detection by CPU

The CPU detects an interrupt request with the highest priority.

8.5.2.5 CPU processing

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

8.5.2.6 Interrupt Service Routine (ISR)

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

(1) Procedure during ISR

An ISR normally pushes register contents to the stack and handles an interrupt as required. The Cortex-M4F core automatically pushes the contents of xPSR, PC, LR, R12 and r3 to r0 to the stack. No extra programming is required for them.

Push the contents of other registers if needed.

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

(2) Clearing an interrupt source

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

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

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

8.6 Exception/Interrupt-Related Registers

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

8.6.1 Register List

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

NVIC registers — Base Address = 0xE000_E000

Register name	Address
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 1	0x0100
Interrupt Set-Enable Register 2	0x0104
Interrupt Set-Enable Register 3	0x0108
Interrupt Clear-Enable Register 1	0x0180
Interrupt Clear-Enable Register 2	0x0184
Interrupt Clear-Enable Register 3	0x0188
Interrupt Set-Pending Register 1	0x0200
Interrupt Set-Pending Register 2	0x0204
Interrupt Set-Pending Register 3	0x0208
Interrupt Clear-Pending Register 1	0x0280
Interrupt Clear-Pending Register 2	0x0284
Interrupt Clear-Pending Register 3	0x0288
Interrupt Priority Register	0x0400 to 0x047F
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18, 0x0D1C, 0x0D20
System Handler Control and State Register	0x0D24

peripheral function name : CG

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

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 was read. Clears on read of any part of the SysTick Control and Status Register.
15-3	−	R	Read as 0.
2	CLKSOURCE	R/W	0: External reference clock (fosc/32) 1: CPU clock (fsys)
1	TICKINT	R/W	0: Do not pend SysTick 1: Pend SysTick
0	ENABLE	R/W	0: Disable 1: Enable If "1" is set, it reloads with the value of the Reload Value Register and starts operation.

Note: In this product, fosc which is selected by CGOSCCR <OSCSEL> by 32 is used as external 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 it to 0. Clearing this register also clears the <COUNTFLAG> bit of the SysTick Control and Status Register.

8.6.2.4 SysTick Calibration Value Register

	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	0	0	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 (Note)

Note: This product does not prepare the calibration value.

8.6.2.5 Interrupt control registers

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

(1) Interrupt Set-Enable Register

This register specifies enabling interrupt and confirms the enable/disable state of interrupt.

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

Writing to "0" is no meaning.

To read this register, be able to confirm the enable/disable state of corresponding interrupt.

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. [95:0] [Write] 1 : Enable interrupt [Read] 0 : The interrupt state is disabled 1 : The interrupt state is enabled

(a) Interrupt Set-Enable Register 1

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

(b) Interrupt Set-Enable Register 2

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

(c) Interrupt Set-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	SETENA (Interrupt 95)	SETENA (Interrupt 94)	SETENA (Interrupt 93)	SETENA (Interrupt 92)	SETENA (Interrupt 91)	SETENA (Interrupt 90)	SETENA (Interrupt 89)	SETENA (Interrupt 88)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA (Interrupt 87)	SETENA (Interrupt 86)	SETENA (Interrupt 85)	SETENA (Interrupt 84)	SETENA (Interrupt 83)	SETENA (Interrupt 82)	SETENA (Interrupt 81)	SETENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA (Interrupt 79)	SETENA (Interrupt 78)	SETENA (Interrupt 77)	SETENA (Interrupt 76)	SETENA (Interrupt 75)	SETENA (Interrupt 74)	SETENA (Interrupt 73)	SETENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA (Interrupt 71)	SETENA (Interrupt 70)	SETENA (Interrupt 69)	SETENA (Interrupt 68)	SETENA (Interrupt 67)	SETENA (Interrupt 66)	SETENA (Interrupt 65)	SETENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

(2) Interrupt Clear-Enable Register

This register specifies disabling interrupt and confirms the enable/disable state of interrupt.

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

Writing to "0" is no meaning.

To read this register, be able to confirm the enable/disable state of corresponding interrupt.

Bit symbol	Type	Function
CLRENA	R/W	Interrupt No. [95:0] [Write] 1 : Disable interrupt [Read] 0 : The interrupt state is disabled 1 : The interrupt state is enabled

(a) Interrupt Clear-Enable Register 1

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

(b) Interrupt Clear-Enable Register 2

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

(c) Interrupt Clear-Enable Register 3

	31	30	29	28	27	26	25	24
bit symbol	CLRENA (Interrupt 95)	CLRENA (Interrupt 94)	CLRENA (Interrupt 93)	CLRENA (Interrupt 92)	CLRENA (Interrupt 91)	CLRENA (Interrupt 90)	CLRENA (Interrupt 89)	CLRENA (Interrupt 88)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA (Interrupt 87)	CLRENA (Interrupt 86)	CLRENA (Interrupt 85)	CLRENA (Interrupt 84)	CLRENA (Interrupt 83)	CLRENA (Interrupt 82)	CLRENA (Interrupt 81)	CLRENA (Interrupt 80)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA (Interrupt 79)	CLRENA (Interrupt 78)	CLRENA (Interrupt 77)	CLRENA (Interrupt 76)	CLRENA (Interrupt 75)	CLRENA (Interrupt 74)	CLRENA (Interrupt 73)	CLRENA (Interrupt 72)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA (Interrupt 71)	CLRENA (Interrupt 70)	CLRENA (Interrupt 69)	CLRENA (Interrupt 68)	CLRENA (Interrupt 67)	CLRENA (Interrupt 66)	CLRENA (Interrupt 65)	CLRENA (Interrupt 64)
After reset	0	0	0	0	0	0	0	0

(3) Interrupt Set-Pending Register

This register specifies pending interrupt and confirms the pending state of interrupt.

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

Writing to "0" is no meaning.

To read this register, be able to confirm the pending state of corresponding interrupt.

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

Bit symbol	Type	Function
SETPEND	R/W	Interrupt No. [95:0] [Write] 1 : Pending interrupt [Read] 0 : No pending 1 : Pending

(a) Interrupt Set-Pending Register 1

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

(b) Interrupt Set-Pending Register 2

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

(c) Interrupt Set-Pending Register 3

	31	30	29	28	27	26	25	24
bit symbol	SETPEND (Interrupt 95)	SETPEND (Interrupt 94)	SETPEND (Interrupt 93)	SETPEND (Interrupt 92)	SETPEND (Interrupt 91)	SETPEND (Interrupt 90)	SETPEND (Interrupt 89)	SETPEND (Interrupt 88)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (Interrupt 87)	SETPEND (Interrupt 86)	SETPEND (Interrupt 85)	SETPEND (Interrupt 84)	SETPEND (Interrupt 83)	SETPEND (Interrupt 82)	SETPEND (Interrupt 81)	SETPEND (Interrupt 80)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (Interrupt 79)	SETPEND (Interrupt 78)	SETPEND (Interrupt 77)	SETPEND (Interrupt 76)	SETPEND (Interrupt 75)	SETPEND (Interrupt 74)	SETPEND (Interrupt 73)	SETPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (Interrupt 71)	SETPEND (Interrupt 70)	SETPEND (Interrupt 69)	SETPEND (Interrupt 68)	SETPEND (Interrupt 67)	SETPEND (Interrupt 66)	SETPEND (Interrupt 65)	SETPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

(4) Interrupt Clear-Pending Register

This register specifies clearing the pended interrupt and confirms the pending state of interrupt.

When set this register to "1", the corresponding pended interrupt is cleared. But this register is invalid for the interrupt which is already started.

Writing to "0" is no meaning.

To read this register, be able to confirm the pending state of corresponding interrupt.

Bit symbol	Type	Function
SETPEND	R/W	Interrupt No. [95:0] [Write] 1 : Clear Pended interrupt [Read] 0 : No pending 1 : Pending

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

(b) Interrupt Clear-Pending Register 2

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

(c) Interrupt Clear-Pending Register 3

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND (Interrupt 95)	CLRPEND (Interrupt 94)	CLRPEND (Interrupt 93)	CLRPEND (Interrupt 92)	CLRPEND (Interrupt 91)	CLRPEND (Interrupt 90)	CLRPEND (Interrupt 89)	CLRPEND (Interrupt 88)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND (Interrupt 87)	CLRPEND (Interrupt 86)	CLRPEND (Interrupt 85)	CLRPEND (Interrupt 84)	CLRPEND (Interrupt 83)	CLRPEND (Interrupt 82)	CLRPEND (Interrupt 81)	CLRPEND (Interrupt 80)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND (Interrupt 79)	CLRPEND (Interrupt 78)	CLRPEND (Interrupt 77)	CLRPEND (Interrupt 76)	CLRPEND (Interrupt 75)	CLRPEND (Interrupt 74)	CLRPEND (Interrupt 73)	CLRPEND (Interrupt 72)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND (Interrupt 71)	CLRPEND (Interrupt 70)	CLRPEND (Interrupt 69)	CLRPEND (Interrupt 68)	CLRPEND (Interrupt 67)	CLRPEND (Interrupt 66)	CLRPEND (Interrupt 65)	CLRPEND (Interrupt 64)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

8.6.2.6 Interrupt Priority Register

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

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

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E434	PRI_55	PRI_54	PRI_53	PRI_52
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E44C	PRI_79	PRI_78	PRI_77	PRI_76
0xE000_E450	PRI_83	PRI_82	PRI_81	PRI_80
0xE000_E454	PRI_87	PRI_86	PRI_85	PRI_84
0xE000_E458	PRI_91	PRI_90	PRI_89	PRI_88
0xE000_E45C	PRI_95	PRI_94	PRI_93	PRI_92

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

The following shows the fields of the Interrupt Priority Registers for interrupt numbers 0 to 3. The Interrupt Priority Registers for all other interrupt numbers have the identical fields. Unused bits return "0" when read, and writing to unused bits has no effect.

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

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

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 top of the space specified in TBLBASE. The offset must be aligned based on the number of exceptions in the table.This means that the minimum alignment is 32 words that you can use for up to 16 interrupts.For more interrupts, you must adjust the alignment by rounding up to the next power of two.
6-0	−	R	Read as 0.

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 0x5FA in the <VECTKEY> field. [Read] Read as 0xFA05.
15	ENDIANESS	R/W	Endianness bit:(Note1) 1: big endian 0: little endian
14-11	−	R	Read as 0.
10-8	PRIGROUP	R/W	Interrupt priority grouping 000: seven bits of pre-emption priority, one bit of 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 to split the interrupt priority register <PRI_n> into pre-emption priority and sub priority.
7-3	−	R	Read as 0.
2	SYSRESET REQ	R/W	System Reset Request. 1=CPU outputs a SYSRESETREQ signal. (note2)
1	VECTCLR ACTIVE	R/W	Clear active vector bit 1: clear all state information for active NMI, fault, and interrupts 0: do not clear. This bit self-clears. It is the responsibility of the application to re initialize the stack.
0	VECTRESET	R/W	System Reset bit 1: reset system 0: do not reset system Resets the system, with the exception of debug components (FPB, DWT and ITM) by setting "1" and this bit is also zero cleared.

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

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

8.6.2.9 System Handler Priority Register

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

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

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

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

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

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

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

8.6.2.10 System Handler Control and State Register

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

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

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

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

8.6.3 Clock generator registers

8.6.3.1 CG Interrupt Mode Control Register

This register specifies the active level to release the low power consumption mode and enable/disable the releasing the low power consumption mode. And detecting 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 101 to 111 : setting prohibited
EMSTx[1:0]	R	Detecting active level (This bit is valid in EMCGx[2:0]="100".) 00: − 01: Rising edge 10: Falling edge 11: Both edges
INTxEN	R/W	Release the low power consumption mode 0 : Disable 1 : Enable

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
INT0	External interrupt pin 0	CGIMCGA <EMCG00[1:0]>	o	o	o	o	o
INT1	External interrupt pin 1	CGIMCGA <EMCG01[1:0]>	o	o	o	o	o
INT2	External interrupt pin 2	CGIMCGA <EMCG02[1:0]>	o	o	o	o	o
INT3	External interrupt pin 3	CGIMCGA <EMCG03[1:0]>	o	o	o	o	o
INT4	External interrupt pin 4	CGIMCGB <EMCG04[1:0]>	o	o	o	o	o
INT5	External interrupt pin 5	CGIMCGB <EMCG05[1:0]>	o	o	o	o	o
INT6	External interrupt pin 6	CGIMCGB <EMCG06[1:0]>	o	o	o	o	o
INT7	External interrupt pin 7	CGIMCGB <EMCG07[1:0]>	o	o	o	o	o
INT8	External interrupt pin 8	CGIMCGC <EMCG08[1:0]>	o	o	o	o	o
INT9	External interrupt pin 9	CGIMCGC <EMCG09[1:0]>	o	o	o	o	o
INTA	External interrupt pin A	CGIMCGC <EMCG0A[1:0]>	o	o	o	o	o
INTB	External interrupt pin B	CGIMCGC <EMCG0B[1:0]>	o	o	o	o	o
INTC	External interrupt pin C	CGIMCGD <EMCG0C[1:0]>	o	o	o	o	o
INTD	External interrupt pin D	CGIMCGD <EMCG0D[1:0]>	o	o	o	o	o
INTE	External interrupt pin E	CGIMCGD <EMCG0E[1:0]>	o	o	o	o	o
INTF	External interrupt pin F	CGIMCGD <EMCG0F[1:0]>	o	o	o	o	o

Note: The active level that is marked with "o" to release the low power consumption mode cab be used.
The active level that is marked with "×" cannot be used.

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

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG03			EMST03		-	INT03EN
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG02			EMST02		-	INT02EN
After reset	0	0	1	0	0	0	undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG01			EMST01		-	INT01EN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG00			EMST00		-	INT00EN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> is depend on the interrupt request. To set correctly, refer to Table 8-4.

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

Note 3: Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bit 31, 23, 15 and 7.

Note 5: Undefined value is read from bit 25, 17, 9 and 1.

(2) CGIMCGB(CG Interrupt Mode Control Register B)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG07			EMST07		-	INT07EN
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG06			EMST06		-	INT06EN
After reset	0	0	1	0	0	0	undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG05			EMST05		-	INT05EN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG04			EMST04		-	INT04EN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> is depend on the interrupt request. To set correctly, refer to Table 8-4.

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

Note 3: Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bit 31, 23, 15 and 7.

Note 5: Undefined value is read from bit 25, 17, 9 and 1.

(3) CGIMCGC(CG Interrupt Mode Control Register C)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG0B			EMST0B		-	INT0BEN
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG0A			EMSTA		-	INT0AEN
After reset	0	0	1	0	0	0	undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG09			EMST9		-	INT09EN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG08			EMST08		-	INT08EN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> is depend on the interrupt request. To set correctly, refer to Table 8-4.

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

Note 3: Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bit 31, 23, 15 and 7.

Note 5: Undefined value is read from bit 25, 17, 9 and 1.

(4) CGIMCGD(CG Interrupt Mode Control Register D)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG0F			EMST0F		-	INT0FEN
After reset	0	0	1	0	0	0	undefined	0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG0E			EMST0E		-	INT0EEN
After reset	0	0	1	0	0	0	undefined	0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG0D			EMST0D		-	INT0DEN
After reset	0	0	1	0	0	0	undefined	0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG0C			EMST0C		-	INT0CEN
After reset	0	0	1	0	0	0	undefined	0

Note 1: The active level specified by <EMCGx[2:0]> is depend on the interrupt request. To set correctly, refer to Table 8-4.

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

Note 3: Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

Note 4: "0" is read from bit 31, 23, 15 and 7.

Note 5: Undefined value is read from bit 25, 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: INT0 0_0100: INT4 0_1000: INT8 0_1100: INTC 0_0001: INT1 0_0101: INT5 0_1001: INT9 0_1101: INTD 0_0010: INT2 0_0110: INT6 0_1010: INTA 0_1110: INTE 0_0011: INT3 0_0111: INT7 0_1011: INTB 0_1111: INTF 1_0000 to 1_1111: Prohibited Read as 0.

8.6.3.3 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 [Read] 0: - 1: Reset from OFD [Write] 0 : Clear reset flag (Note2) 1 : don't care
4	DBGRSTF	R/W	Debug reset flag (Note2) [Read] 0: - 1: Reset from SYSRESETREQ [Write] 0 : Clear reset flag (Note2) 1 : don't care
3	VLTDRSTF	R/W	VLTD reset flag [Read] 0: - 1: Reset from VLTD [Write] 0 : Clear reset flag (Note2) 1 : don't care
2	WDTRSTF	R/W	WDT reset flag [Read] 0: - 1: Reset from WDT [Write] 0 : Clear reset flag (Note2) 1 : don't care
1	PINRSTF	R/W	$\overline{\text{RESET}}$ pin flag [Read] 0: - 1: Reset from $\overline{\text{RESET}}$ pin. [Write] 0 : Clear reset flag (Note2) 1 : don't care
0	PONRSTF	R/W	Power On Reset flag [Read] 0: - 1: Reset from Power on. [Write] 0 : Clear reset flag (Note2) 1 : don't care

Note 1: The reset which is generated by application interrupt in NVIC of CPU and setting reset control register <SYSRESETREQ> is displayed.

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

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

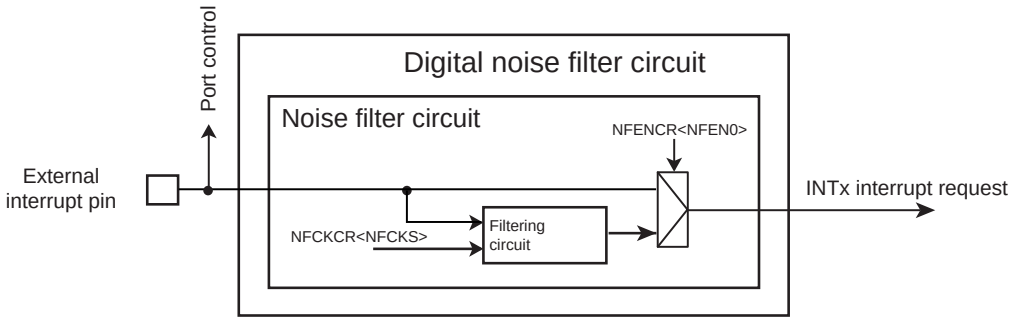


Figure 9-1 Circuit diagram of digital noise filter

9.2 Registers

9.2.1 Register List

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+)
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 1: NFCKCR<NFCKS> setting is specified in NFENCR<NFEN[15:0]>= all "0".

Note 2: 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	NFEN15	NFEN14	NFEN13	NFEN12	NFEN11	NFEN10	NFEN9	NFEN8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	NFEN7	NFEN6	NFEN5	NFEN4	NFEN3	NFEN2	NFEN1	NFEN0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15	NFEN15	R/W	INTF 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)
14	NFEN14	R/W	INTE 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)
13	NFEN13	R/W	INTD 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)
12	NFEN12	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)
11	NFEN11	R/W	INTB 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)
10	NFEN10	R/W	INTA 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)
9	NFEN9	R/W	INT9 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)
8	NFEN8	R/W	INT8 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)
7	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)
6	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)

Bit	Bit Symbol	Type	Function
5	NFEN5	R/W	INT5 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)
4	NFEN4	R/W	INT4 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)
3	NFEN3	R/W	INT3 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)
2	NFEN2	R/W	INT2 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	NFEN1	R/W	INT1 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	NFEN0	R/W	INT0 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 1: 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 2: Before external interrupt signals are enabled, clear the interrupt events and then set the corresponding bit of NFENCR register to be enabled.

Note 3: 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. µDMA Controller (µDMAC)

10.1 Overview

10.1.1 Function List

The main functions per unit are shown as below:

For the information on the start trigger of peripheral functions, refer to the chapter on "Product Information."

Table 10-1 µDMA outline (Per unit)

Functions	Features		Descriptions
Channels	32 channels		-
Start trigger	Start by Hardware		DMA requests from peripheral functions
	Start by Software		Specified by the DMAxChnlSwRequest register
Priority	Between channels	ch0 (high priority) > … > ch31 (high priority) > ch0 (Normal priority) > … > ch31 (Normal priority)	High-priority can be configured by the DMAxChnl-PrioritySet register
Transfer data size	8/16/32 bits		-
The number of transfer	1 to 1024 times		-
Address	Transfer source address	Increment / fixed	Transfer source address and destination address can be selected from the increment setting or fixed setting.
	Transfer destination address	Increment / fixed	
Endian	Little-endian		-
Interrupt function	Transfer completion interrupt		Output for each unit
	Error interrupt		
Operation mode	Basic mode Automatic request mode Ping-pong mode Memory scatter/gather mode Peripheral scatter/gather mode		-

10.2 Block Diagram

The µDMA controller contains the following blocks:

- APB block

  This block controls the access to the control register.
- AHB block

  This block controls the bus cycle of the DMA transfer.
- DMA control block

  This block controls the whole operation of the DMA.
- Interrupt control block

  This block integrates interrupts and sets the flag register.

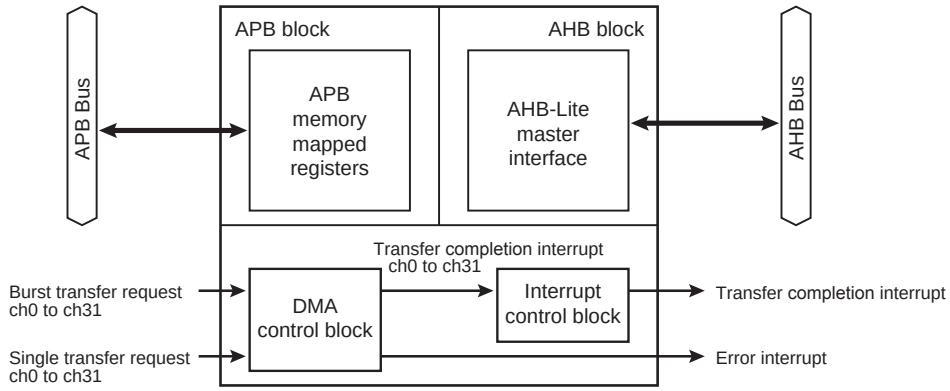


Figure 10-1 µDMA block diagram

10.3 Registers

10.3.1 Register List

The following table shows control registers and addresses:

For the base address, refer to "A list of peripheral function base addresses" in the chapter on "Memory Map."

Then name of peripheral: DMA

Register names		Address (Base+)
DMA status register	DMAxStatus	0x0000
DMA configuration register	DMAxCfg	0x0004
Channel control data base pointer register	DMAxCtrlBasePtr	0x0008
Channel alternate control data base pointer register	DMAxAltCtlBasePtr	0x000C
Channel software request status register	DMAxChnlSwRequest	0x0014
Channel useburst set register	DMAxChnlUseburstSet	0x0018
Channel useburst clear register	DMAxChnlUseburstClr	0x001C
Channel request mask set register	DMAxChnlReqMaskSet	0x0020
Channel request mask clear register	DMAxChnlReqMaskClr	0x0024
Channel enable set register	DMAxChnlEnableSet	0x0028
Channel enable clear register	DMAxChnlEnableClr	0x002C
Channel primary-alternate set register	DMAxChnlPriAltSet	0x0030
Channel primary-alternate clear register	DMAxChnlPriAltClr	0x0034
Channel priority set register	DMAxChnlPrioritySet	0x0038
Channel priority clear register	DMAxChnlPriorityClr	0x003C
Bus error clear register	DMAxErrClr	0x004C

Then name of peripheral: DMAIF

Register name		Address (Base+)
Flag register A	DMAIFFLGA	0x0000

Note: Access the registers in units of words (32 bits).

10.3.2 DMAxStatus (DMAC Status Register)

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

Bit	Bit symbol	Type	Functions
31-29	-	R	Read as "0".
28	-	R	Read as "1".
27-21	-	R	Read as "0".
20-16	-	R	Read as "1".
15-8	-	R	Read as "0".
7-4	-	R	Read as an undefined value.
3-1	-	R	Read as "0".
0	master_enable	R	DMA operation 0: Disabled 1: Enabled

10.3.3 DMAxCfg (DMAC Configuration Register)

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

Bit	Bit symbol	Type	Function
31-1	-	W	Write as "0".
0	master_ enable	W	DMA operation 0: Disabled 1: Enabled

Note: After DMAxCfg = 0x00000001, DMAxChnlReqMaskSet = 0xFFFFFFFF and DMAxChnlEnableSet = 0xFFFFFFFF are set, set "1" to the corresponding bit of DMAxChanlReqMaskClr to release masking of the channel to be used.

10.3.4 DMAxCtrlBasePtr (Channel Control Data Base-pointer Register)

	31	30	29	28	27	26	25	24
Bit symbol	ctrl_base_ptr							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	ctrl_base_ptr							
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	ctrl_base_ptr						-	-
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
31-10	ctrl_base_ptr	R/W	Primary data base-pointer Specifies the base address of the primary data.
9-0	-	R	Read as "0".

10.3.5 DMAxAltCtrlBasePtr (Channel Alternate Control Data Base-pointer Register)

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

Bit	Bit symbol	Type	Function
31-0	alt_ctrl_base_pt	R	Alternative data base-pointer. Reads the base address of the alternative data.

10.3.6 DMAxChnlSwRequest (Channel Software Request Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_sw_request (ch31)	chnl_sw_request (ch30)	chnl_sw_request (ch29)	chnl_sw_request (ch28)	chnl_sw_request (ch27)	chnl_sw_request (ch26)	chnl_sw_request (ch25)	chnl_sw_request (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	chnl_sw_request (ch23)	chnl_sw_request (ch22)	chnl_sw_request (ch21)	chnl_sw_request (ch20)	chnl_sw_request (ch19)	chnl_sw_request (ch18)	chnl_sw_request (ch17)	chnl_sw_request (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	chnl_sw_request (ch15)	chnl_sw_request (ch14)	chnl_sw_request (ch13)	chnl_sw_request (ch12q)	chnl_sw_request (ch11)	chnl_sw_request (ch10)	chnl_sw_request (ch9)	chnl_sw_request (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	chnl_sw_request (ch7)	chnl_sw_request (ch6)	chnl_sw_request (ch5)	chnl_sw_request (ch4)	chnl_sw_request (ch3)	chnl_sw_request (ch2)	chnl_sw_request (ch1)	chnl_sw_request (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-0	chnl_sw_request	W	DMA request 0: A transfer request does not occur. 1: A transfer request occurs. Specifies transfer requests to the each channel.

10.3.7 DMAxChnlUseburstSet (Channel useburst Set Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_useburst_set (ch31)	chnl_useburst_set (ch30)	chnl_useburst_set (ch29)	chnl_useburst_set (ch28)	chnl_useburst_set (ch27)	chnl_useburst_set (ch26)	chnl_useburst_set (ch25)	chnl_useburst_set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	chnl_useburst_set (ch23)	chnl_useburst_set (ch22)	chnl_useburst_set (ch21)	chnl_useburst_set (ch20)	chnl_useburst_set (ch19)	chnl_useburst_set (ch18)	chnl_useburst_set (ch17)	chnl_useburst_set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	chnl_useburst_set (ch15)	chnl_useburst_set (ch14)	chnl_useburst_set (ch13)	chnl_useburst_set (ch12)	chnl_useburst_set (ch11)	chnl_useburst_set (ch10)	chnl_useburst_set (ch9)	chnl_useburst_set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	chnl_useburst_set (ch7)	chnl_useburst_set (ch6)	chnl_useburst_set (ch5)	chnl_useburst_set (ch4)	chnl_useburst_set (ch3)	chnl_useburst_set (ch2)	chnl_useburst_set (ch1)	chnl_useburst_set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-0	chnl_useburst_set	R/W	Single-transfer is disabled [Write] 1: Single-transfer is disabled. [Read] 0: Single-transfer is enabled. 1: Single-transfer is disabled. Each bit corresponds to the channels in the specified number. Writing "1" disables the single-transfer to the corresponding channel, and only burst transfer request becomes valid. Writing "0" has no meaning. Set the DMAxChnlUseburstClr register in order to cancel the disabled the single-transfer. By reading the bit, the channel state of the corresponding bit can be checked whether it is enabled or disabled. Bits are automatically set in the following conditions: ・ This bit is cleared to "0", if the number of remaining transfer is less than 2^R times at the end of second 2^R time transfer from the end ("R" is specified by the channel_cfg<R_power> of the control data). ・ If the channel_cfg<next_useburst> of the control data is set to "1" in the peripheral scatter/gather mode, this bit is set to "1" when the DMA transfer of the alternative data ends.

Note: Do not set this bit to "1" if you do not use the burst transfer request on the condition where the number of transfers is less than 2^R times.

10.3.8 DMAxChnlUseburstClr (Channel useburst Clear Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_useburst_clr (ch31)	chnl_useburst_clr (ch30)	chnl_useburst_clr (ch29)	chnl_useburst_clr (ch28)	chnl_useburst_clr (ch27)	chnl_useburst_clr (ch26)	chnl_useburst_clr (ch25)	chnl_useburst_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	chnl_useburst_clr (ch23)	chnl_useburst_clr (ch22)	chnl_useburst_clr (ch21)	chnl_useburst_clr (ch20)	chnl_useburst_clr (ch19)	chnl_useburst_clr (ch18)	chnl_useburst_clr (ch17)	chnl_useburst_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	chnl_useburst_clr (ch15)	chnl_useburst_clr (ch14)	chnl_useburst_clr (ch13)	chnl_useburst_clr (ch12)	chnl_useburst_clr (ch11)	chnl_useburst_clr (ch10)	chnl_useburst_clr (ch9)	chnl_useburst_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	chnl_useburst_clr (ch7)	chnl_useburst_clr (ch6)	chnl_useburst_clr (ch5)	chnl_useburst_clr (ch4)	chnl_useburst_clr (ch3)	chnl_useburst_clr (ch2)	chnl_useburst_clr (ch1)	chnl_useburst_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-0	chnl_useburst_clr	W	Single-transfer is enabled. 1: Enables the single-transfer. Each bit corresponds to the channels in the specified number. Writing "1" enables the single-transfer to the corresponding channel. Writing "0" has no meaning. To disable or confirm the signal-transfer, configure the DMAxChnlUseburstSet register.

10.3.9 DMAxChnlReqMaskSet (Channel Request Mask Set Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_req_mask_set (ch31)	chnl_req_mask_set (ch30)	chnl_req_mask_set (ch29)	chnl_req_mask_set (ch28)	chnl_req_mask_set (ch27)	chnl_req_mask_set (ch26)	chnl_req_mask_set (ch25)	chnl_req_mask_set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	chnl_req_mask_set (ch23)	chnl_req_mask_set (ch22)	chnl_req_mask_set (ch21)	chnl_req_mask_set (ch20)	chnl_req_mask_set (ch19)	chnl_req_mask_set (ch18)	chnl_req_mask_set (ch17)	chnl_req_mask_set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	chnl_req_mask_set (ch15)	chnl_req_mask_set (ch14)	chnl_req_mask_set (ch13)	chnl_req_mask_set (ch12)	chnl_req_mask_set (ch11)	chnl_req_mask_set (ch10)	chnl_req_mask_set (ch9)	chnl_req_mask_set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	chnl_req_mask_set (ch7)	chnl_req_mask_set (ch6)	chnl_req_mask_set (ch5)	chnl_req_mask_set (ch4)	chnl_req_mask_set (ch3)	chnl_req_mask_set (ch2)	chnl_req_mask_set (ch1)	chnl_req_mask_set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-0	chnl_req_mask_set	R/W	DMA request masking [Write] 1: Mask a DMA request [Read] 0: A DMA request is valid. 1: A DMA request is invalid. Each bit corresponds to the channels in the specified number. Writing "1" disables the single-transfer for the corresponding channel. Writing "0" has no meaning. To disable masking, configure the DMAxChnlReqMaskClr register. By reading the bit, the status of the DMA request setting can be checked whether it is enabled or disabled.

10.3.10 DMAxChnlReqMaskClr (Channel Request Mask Clear Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_req_mask_clr (ch31)	chnl_req_mask_clr (ch30)	chnl_req_mask_clr (ch29)	chnl_req_mask_clr (ch28)	chnl_req_mask_clr (ch27)	chnl_req_mask_clr (ch26)	chnl_req_mask_clr (ch25)	chnl_req_mask_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	chnl_req_mask_clr (ch23)	chnl_req_mask_clr (ch22)	chnl_req_mask_clr (ch21)	chnl_req_mask_clr (ch20)	chnl_req_mask_clr (ch19)	chnl_req_mask_clr (ch18)	chnl_req_mask_clr (ch17)	chnl_req_mask_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	chnl_req_mask_clr (ch15)	chnl_req_mask_clr (ch14)	chnl_req_mask_clr (ch13)	chnl_req_mask_clr (ch12)	chnl_req_mask_clr (ch11)	chnl_req_mask_clr (ch10)	chnl_req_mask_clr (ch9)	chnl_req_mask_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	chnl_req_mask_clr (ch7)	chnl_req_mask_clr (ch6)	chnl_req_mask_clr (ch5)	chnl_req_mask_clr (ch4)	chnl_req_mask_clr (ch3)	chnl_req_mask_clr (ch2)	chnl_req_mask_clr (ch1)	chnl_req_mask_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-0	chnl_req_mask_clr	W	DMA request mask clear 1: Clears the corresponding channel of the DMA request mask. Each bit corresponds to the channels in the specified number. Writing "1" disables the DMA request mask setting of the corresponding channel. Writing "0" has no meaning. Configure the DMAxChnlReqMaskSet register to enable and confirm the setting.

10.3.11 DMAxChnlEnableSet (Channel Enable Set Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_enable_set (ch31)	chnl_enable_set (ch30)	chnl_enable_set (ch29)	chnl_enable_set (ch28)	chnl_enable_set (ch27)	chnl_enable_set (ch26)	chnl_enable_set (ch25)	chnl_enable_set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	chnl_enable_set (ch23)	chnl_enable_set (ch22)	chnl_enable_set (ch21)	chnl_enable_set (ch20)	chnl_enable_set (ch19)	chnl_enable_set (ch18)	chnl_enable_set (ch17)	chnl_enable_set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	chnl_enable_set (ch15)	chnl_enable_set (ch14)	chnl_enable_set (ch13)	chnl_enable_set (ch12)	chnl_enable_set (ch11)	chnl_enable_set (ch10)	chnl_enable_set (ch9)	chnl_enable_set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	chnl_enable_set (ch7)	chnl_enable_set (ch6)	chnl_enable_set (ch5)	chnl_enable_set (ch4)	chnl_enable_set (ch3)	chnl_enable_set (ch2)	chnl_enable_set (ch1)	chnl_enable_set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-0	chnl_enable_set	R/W	DMA operation [Write] 1: Enable the corresponding channel. [Read] 0: The corresponding bit is invalid. 1: The corresponding bit is valid. Each bit corresponds to the channels in the specified number. Writing "1" enables the corresponding channels. Writing "0" has no meaning. To disable the setting, configure the DMAxChnlEnableClr register. By reading the bit, the corresponding channel can be checked whether it is enabled or disabled. In the following conditions, the function automatically becomes invalid. ・DMA cycle ends. ・If the channel_cfg<cycle_ctrl> reads the control data of "000". ・A bus error occurs.

10.3.12 DMAxChnlEnableClr (Channel Enable Clear Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_enable_clr (ch31)	chnl_enable_clr (ch30)	chnl_enable_clr (ch29)	chnl_enable_clr (ch28)	chnl_enable_clr (ch27)	chnl_enable_clr (ch26)	chnl_enable_clr (ch25)	chnl_enable_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	chnl_enable_clr (ch23)	chnl_enable_clr (ch22)	chnl_enable_clr (ch21)	chnl_enable_clr (ch20)	chnl_enable_clr (ch19)	chnl_enable_clr (ch18)	chnl_enable_clr (ch17)	chnl_enable_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	chnl_enable_clr (ch15)	chnl_enable_clr (ch14)	chnl_enable_clr (ch13)	chnl_enable_clr (ch12)	chnl_enable_clr (ch11)	chnl_enable_clr (ch10)	chnl_enable_clr (ch9)	chnl_enable_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	chnl_enable_clr (ch7)	chnl_enable_clr (ch6)	chnl_enable_clr (ch5)	chnl_enable_clr (ch4)	chnl_enable_clr (ch3)	chnl_enable_clr (ch2)	chnl_enable_clr (ch1)	chnl_enable_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-0	chnl_enable_clr	W	DMA disabled 1: Disables the corresponding channel. Each bit corresponds to the channels in the specified number. Writing "1" disables the corresponding channel. Writing "0" has no meaning. Configure the DMAxChnlEnableSet register in order to enable and confirm the setting. In the following conditions, the function automatically becomes invalid. ・ DMA cycle ends. ・ The channel_cfg<cycle_ctrl> reads the control data of "000". ・ A bus error occurs.

10.3.13 DMAxChnlPriAltSet (Channel Primary-alternate Set Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_pri_alt_set (ch31)	chnl_pri_alt_set (ch30)	chnl_pri_alt_set (ch29)	chnl_pri_alt_set (ch28)	chnl_pri_alt_set (ch27)	chnl_pri_alt_set (ch26)	chnl_pri_alt_set (ch25)	chnl_pri_alt_set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	chnl_pri_alt_set (ch23)	chnl_pri_alt_set (ch22)	chnl_pri_alt_set (ch21)	chnl_pri_alt_set (ch20)	chnl_pri_alt_set (ch19)	chnl_pri_alt_set (ch18)	chnl_pri_alt_set (ch17)	chnl_pri_alt_set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	chnl_pri_alt_set (ch15)	chnl_pri_alt_set (ch14)	chnl_pri_alt_set (ch13)	chnl_pri_alt_set (ch12)	chnl_pri_alt_set (ch11)	chnl_pri_alt_set (ch10)	chnl_pri_alt_set (ch9)	chnl_pri_alt_set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	chnl_pri_alt_set (ch7)	chnl_pri_alt_set (ch6)	chnl_pri_alt_set (ch5)	chnl_pri_alt_set (ch4)	chnl_pri_alt_set (ch3)	chnl_pri_alt_set (ch2)	chnl_pri_alt_set (ch1)	chnl_pri_alt_set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-0	chnl_pri_alt_set	R/W	Selects primary data or alternative data [Write] 1: Uses alternative data [Read] 0: Primary data 1: Alternative data Each bit corresponds to the channels in the specified number. Writing "1" specifies the data that is firstly used in the corresponding channel as "alternative data". Writing "0" has no meaning. To disable this bit, use the DMAxChnlEnableClr register. Only in basic mode, automatic request mode, and ping-pong mode the first data can be specified as alternative data. When this bit is read, data of the corresponding channel can be checked whether data is primary data or alternative data. In the following conditions, the settings are automatically changed. ・ The primary data transfer is completed in ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode. ・ Data transfer of the alternative data is completed in the ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode.

10.3.14 DMAxChnlPriAltClr (Channel Primary-alternate Clear Register)

	31	30	29	28	27	26	25	24
Bit symbol	chn_pri_alt_clr (ch31)	chn_pri_alt_clr (ch30)	chn_pri_alt_clr (ch29)	chn_pri_alt_clr (ch28)	chn_pri_alt_clr (ch27)	chn_pri_alt_clr (ch26)	chn_pri_alt_clr (ch25)	chn_pri_alt_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	chn_pri_alt_clr (ch23)	chn_pri_alt_clr (ch22)	chn_pri_alt_clr (ch21)	chn_pri_alt_clr (ch20)	chn_pri_alt_clr (ch19)	chn_pri_alt_clr (ch18)	chn_pri_alt_clr (ch17)	chn_pri_alt_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	chn_pri_alt_clr (ch15)	chn_pri_alt_clr (ch14)	chn_pri_alt_clr (ch13)	chn_pri_alt_clr (ch12)	chn_pri_alt_clr (ch11)	chn_pri_alt_clr (ch10)	chn_pri_alt_clr (ch9)	chn_pri_alt_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	chn_pri_alt_clr (ch7)	chn_pri_alt_clr (ch6)	chn_pri_alt_clr (ch5)	chn_pri_alt_clr (ch4)	chn_pri_alt_clr (ch3)	chn_pri_alt_clr (ch2)	chn_pri_alt_clr (ch1)	chn_pri_alt_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-0	chnl_pri_alt_clr	W	Clears the alternative data setting. 1: Uses the primary data Each bit corresponds to the channels in the specified number. Writing "1" sets the data of the corresponding channel to the primary data. Setting "0" is invalid. Configure the DMAxChnlPriAltSet register to set the primary data or to confirm the setting. In the following conditions, the setting is automatically changed. ・ The primary data transfer in memory scatter/gather mode or peripheral scatter/gather mode is complete. ・ The primary data transfer in ping-pong mode is complete. ・ The alternative transfer in ping-pong mode, memory scatter/gather mode, or peripheral scatter/gather mode is complete.

10.3.15 DMAxChnlPrioritySet (Channel Priority Set Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_priority_set (ch31)	chnl_priority_set (ch30)	chnl_priority_set (ch29)	chnl_priority_set (ch28)	chnl_priority_set (ch27)	chnl_priority_set (ch26)	chnl_priority_set (ch25)	chnl_priority_set (ch24)
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	chnl_priority_set (ch23)	chnl_priority_set (ch22)	chnl_priority_set (ch21)	chnl_priority_set (ch20)	chnl_priority_set (ch19)	chnl_priority_set (ch18)	chnl_priority_set (ch17)	chnl_priority_set (ch16)
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	chnl_priority_set (ch15)	chnl_priority_set (ch14)	chnl_priority_set (ch13)	chnl_priority_set (ch12)	chnl_priority_set (ch11)	chnl_priority_set (ch10)	chnl_priority_set (ch9)	chnl_priority_set (ch8)
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	chnl_priority_set (ch7)	chnl_priority_set (ch6)	chnl_priority_set (ch5)	chnl_priority_set (ch4)	chnl_priority_set (ch3)	chnl_priority_set (ch2)	chnl_priority_set (ch1)	chnl_priority_set (ch0)
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-0	chnl_priority_set	R/W	Priority settings [Write] 1: Sets the high-priority [Read] 0: Normal priority 1: High priority Each bit corresponds to the channels in the specified number. Writing "1" sets the priority of the corresponding channel high. Writing "0" has no meaning.To change the priority again to the normal, configure the DMAxChnlPriorityClr register. the priority of the corresponding channel, high-priority or normal priority, can be confirmed by reading the bit.

10.3.16 DMAxChnlPriorityClr (Channel Priority Clear Register)

	31	30	29	28	27	26	25	24
Bit symbol	chnl_priority_clr (ch31)	chnl_priority_clr (ch30)	chnl_priority_clr (ch29)	chnl_priority_clr (ch28)	chnl_priority_clr (ch27)	chnl_priority_clr (ch26)	chnl_priority_clr (ch25)	chnl_priority_clr (ch24)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	23	22	21	20	19	18	17	16
Bit symbol	chnl_priority_clr (ch23)	chnl_priority_clr (ch22)	chnl_priority_clr (ch21)	chnl_priority_clr (ch20)	chnl_priority_clr (ch19)	chnl_priority_clr (ch18)	chnl_priority_clr (ch17)	chnl_priority_clr (ch16)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	15	14	13	12	11	10	9	8
Bit symbol	chnl_priority_clr (ch15)	chnl_priority_clr (ch14)	chnl_priority_clr (ch13)	chnl_priority_clr (ch12)	chnl_priority_clr (ch11)	chnl_priority_clr (ch10)	chnl_priority_clr (ch9)	chnl_priority_clr (ch8)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
	7	6	5	4	3	2	1	0
Bit symbol	chnl_priority_clr (ch7)	chnl_priority_clr (ch6)	chnl_priority_clr (ch5)	chnl_priority_clr (ch4)	chnl_priority_clr (ch3)	chnl_priority_clr (ch2)	chnl_priority_clr (ch1)	chnl_priority_clr (ch0)
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Bit	Bit symbol	Type	Function
31-0	chnl_priority_clr	W	Clears the high-priority setting. [Write] 1:Sets normal priority setting Each bit corresponds to the channels in the specified number. Writing "1" changes the priority of the corresponding channel to normal priority. Writing "0" has no meaning. Configure the DMAxChnlPrioritySet register to set the high-priority and to confirm the setting.

10.3.17 DMAxErrClr (Bus Error 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	-	-	-	-	-	-	-	err_clr
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-1	-	R	Read as "0".
0	err_clr	R/W	Bus error [Write] 1: Clears a bus error. [Read] 0: No bus error 1: The state of a bus error A bus error occurrence can be confirmed by reading the bit. Writing "1" clears a bus error. Writing "0" has no meaning.

10.3.18 DMAIFFLGx (DMA Flag Register)

	31	30	29	28	27	26	25	24
Bit symbol	FLG31	FLG30	FLG29	FLG28	FLG27	FLG26	FLG25	FLG24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	FLG23	FLG22	FLG21	FLG20	FLG19	FLG18	FLG17	FLG16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	FLG15	FLG14	FLG13	FLG12	FLG11	FLG10	FLG9	FLG8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	FLG7	FLG6	FLG5	FLG4	FLG3	FLG2	FLG1	FLG0
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Functions
31-0	FLG31 - FLG0	R	DMA factor flag 0:- 1: A completion interrupt occurs. A bit number corresponds with a channel number. If a transfer completion interrupt occurs, the corresponding bit is set to "1". This register is cleared by reading automatically.

Note: Before a transfer completion interrupt is enabled, this register must be read and cleared.

10.4 Operation

This DMA is controlled by the channel control data, which locates on the memory. A channel of the each data is four words and allocated in the contiguous areas same as the number of channels.

There are two types of channel control data: primary data and alternative data. According to the operation mode, one of them is selected by setting the register or both data is used.

10.4.1 Channel Control Data Memory Map

Figure 10-2 shows the example of memory map of the channel control data.

Set the start address of the primary data to the DMAxCtrlBasePtr and the start address of the alternative data to the DMAxAltCtrlBasePtr.

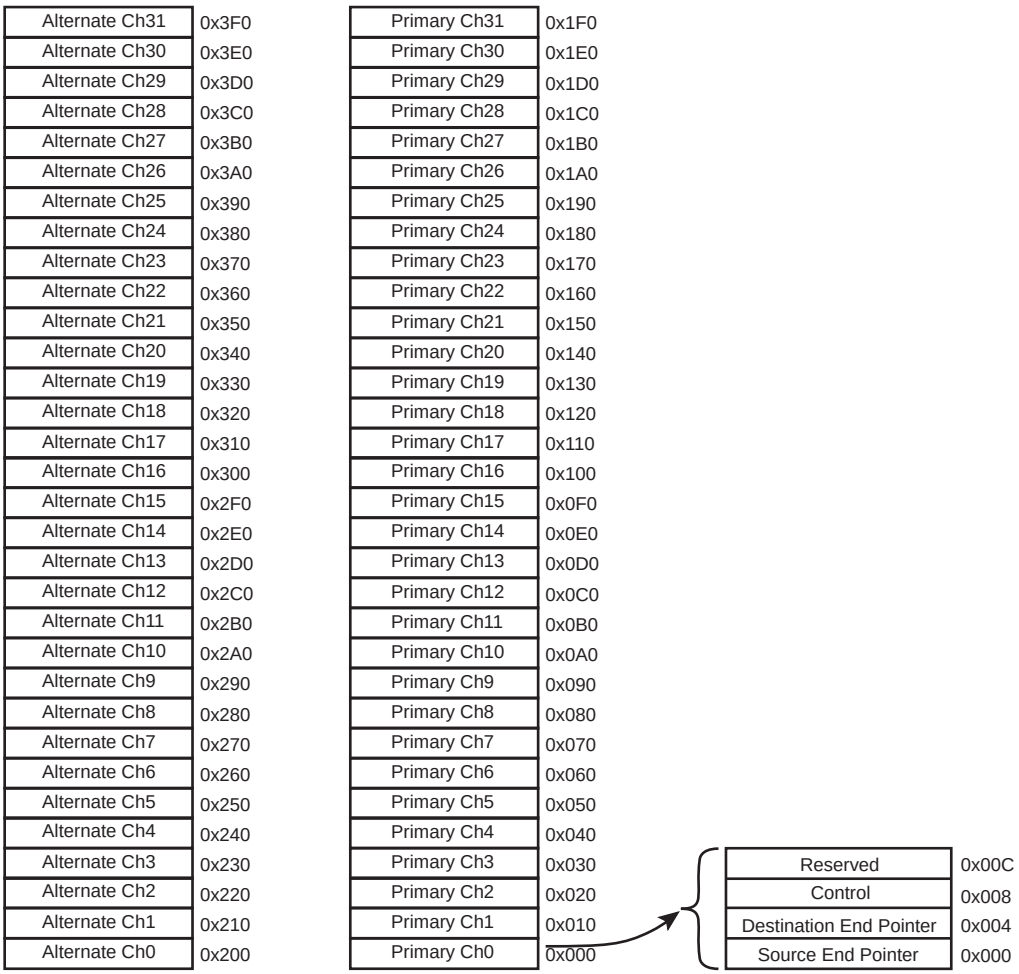


Figure 10-2 Memory map of the control data

Figure 10-2 shows the memory map of which all 32 channels can be used. Necessary areas are determined by the number of usable channels. Table 10-2 shows the relationship between the number of channels and addresses.

Table 10-2 Address bit setting of channel control

Channel	Address							Settable base address
	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]	
0	-	-	-	-	-	A	Channel control data setting	0xXXXX_XX00, 0xXXXX_XX20, 0xXXXX_XX40, 0xXXXX_XX60, 0xXXXX_XX80, 0xXXXX_XXA0, 0xXXXX_XXC0, 0xXXXX_XXE0
0 to 1	-	-	-	-	A	C[0]		0xXXXX_XX00, 0xXXXX_XX40, 0xXXXX_XX80, 0xXXXX_XXC0
0 to 3	-	-	-	A	C[1:0]			0xXXXX_XX00, 0xXXXX_XX80
0 to 7	-	-	A	C[2:0]				0xXXXX_X000, 0xXXXX_X100, 0xXXXX_X200, 0xXXXX_X300, 0xXXXX_X400, 0xXXXX_X500, 0xXXXX_X600, 0xXXXX_X700, 0xXXXX_X800, 0xXXXX_X900, 0xXXXX_XA00, 0xXXXX_XB00, 0xXXXX_XC00, 0xXXXX_XD00, 0xXXXX_XE00, 0xXXXX_XF00
0 to 15	-	A	C[3:0]					0xXXXX_X000, 0xXXXX_X200, 0xXXXX_X400, 0xXXXX_X600, 0xXXXX_X800, 0xXXXX_XA00, 0xXXXX_XC00, 0xXXXX_XE00
0 to 31	A	C[4:0]						0xXXXX_X000, 0xXXXX_X400, 0xXXXX_X800, 0xXXXX_XC00

A: Primary/alternative setting (0:primary, 1:alternative)
C[x:0]: Channel number setting

10.4.2 Channel Control Data Structure

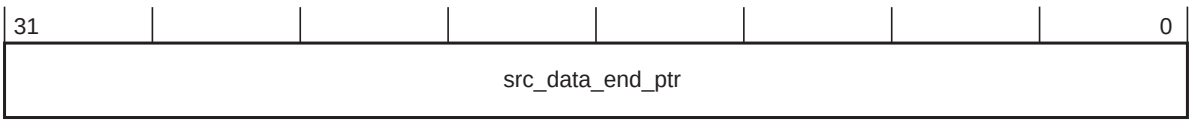
The channel control data contains the three kinds of data shown below:

- The final address of the transfer source address
- The final address of the transfer destination address
- Control data

Each data is described in the following sections:

10.4.2.1 Final Address of the Transfer Source Data

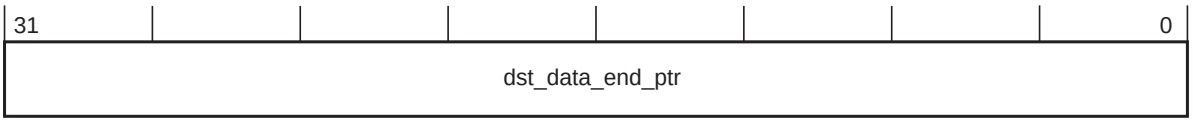
Specify the final address of the data to be transferred. The alignment of an address should be adjusted to a transfer data size. The DMA calculates the start address of the source address using this data.



bit	Bit symbol	Function
[31:0]	src_data_end_ptr	The final address of source transfer data

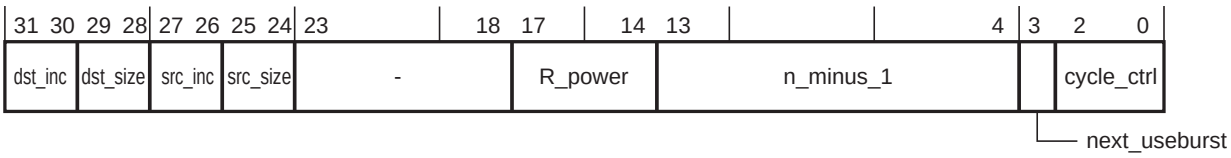
10.4.2.2 Final Address of the Transfer Destination Address

Specify the final address of the destination address. The alignment of an address should be adjusted to a transfer data size. The DMA calculates the start address of the destination address of the transfer destination address.



bit	Bit symbol	Function
[31:0]	dst_data_end_ptr	The final address of the transfer destination address.

10.4.2.3 Control Data Setting



bit	Bit symbol	Function
[31:30]	dst_inc	Increments the transfer destination address (note 2) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: No increment
[29:28]	dst_size	Data size of transfer destination (note1) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: Reserved
[27:26]	src_inc	Increments the transfer source address (note 2) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: No increment
[25:24]	src_size	Data size of transfer source (note 1) 00: 1 byte 01: 2 bytes 10: 4 bytes 11: Reserved
[23:18]	-	Set "000000".

bit	Bit symbol	Function
[17:14]	R_power	Arbitration 0000: After 1 transfer 0001: After 2 transfers 0010: After 4 transfers 0011: After 8 transfers 0100: After 16 transfers 0101: After 32 transfers 0110: After 64 transfers 0111: After 128 transfers 1000: After 256 transfers 1001: After 512 transfers 1010 - 1111: No arbitration A transfer request is checked after the specified number of transfers. If a higher-priority request exists, the controller arbitrates the DMA transfer.
[13:4]	n_minus_1	The number of transfers 0x000: Once 0x001: Twice 0x002: Three times : 0x3FF: 1024 times
[3]	next_useburst	Changes the setting of single-transfer 0: Do not change the value of <chnl_useburst_set>. 1: Sets <chnl_useburst_set> to "1". Specifies whether to set "1" to the <chnl_useburst_set> bit at the end of the DMA transfer using alternative data in the peripheral scatter/ gather mode. Note) This bit <chnl_useburst_set> is zero cleared, if the number of remaining transfer is less than 2^R times at the end of second 2^Rtime transfer from the end ("R" is specified by the <R_power>). Setting this bit to "1" sets "1" to the <chnl_userburst_set>.
[2:0]	cycle_ctrl	Operation mode 000: Invalid. The DMA stops the operation. 001: Basic mode 010: Automatic request mode 011: Ping-pong mode 100: Memory scatter / gather mode (primary data) 101: Memory scatter / gather mode (alternative data) 110: Peripheral memory scatter / gather mode (primary data) 111: Peripheral memory scatter / gather mode (alternative data)

Note 1: The setting value of <dst_size> must be the same as <src_size>.

Note 2: According to the settings of <dst_size> and <src_size>, the settings of <dst_inc> and <src_inc> are limited as shown below:

<src_inc>/<dst_inc>	<src_size>/<dst_size>		
	00 (1 byte)	01 (2 bytes)	10 (4 bytes)
00 (1byte)	o	-	-
01 (2bytes)	o	o	-
10 (4bytes)	o	o	o
No increment	o	o	o

10.4.3 Operation Modes

This section describes the operation modes configured by channel_cfg<cycle_ctrl> of the channel control data.

10.4.3.1 Invalid Setting

The DMA sets the operation mode invalid after the end of transfer. This operation prevents a transfer from being performed again. Also, the operation completes if invalid data is read either in ping-pong mode, memory scatter / gather mode or peripheral scatter / gather mode.

10.4.3.2 Basic Mode

In basic mode, data structure can be selected from primary data or alternative data.

A transfer is started by receiving a transfer request.

An arbitration is performed for every transfer configured by <R_power>. If a higher-priority request exits, the DMA switches a channel. If a transfer request for the operating channel is received, the transfer is continued.

After performing transfers for the number of times specified by <n_minus_1>, a transfer completion interrupt occurs.

10.4.3.3 Automatic Request Mode

In this mode, a single-transfer request stops the DMA transfer. The data structure can be selected from primary data or alternative data.

The DMA transfer is started by a transfer request.

In each transfer configured by <R_power>, a channel is switched if a higher-priority request is received. If not, the transfer is continued.

After performing transfers for the number of times specified by <n_minus_1>, a transfer completion interrupt occurs.

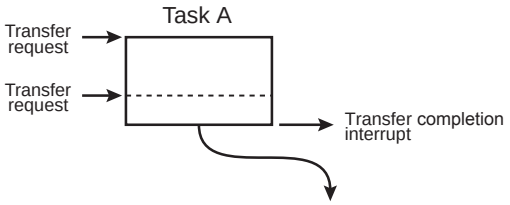
10.4.3.4 Ping-pong Mode

In ping-pong mode, a continuous DMA transfer that uses primary data and alternative data alternately is performed. If <cycle_ctrl> reads data specified to be invalid ("000"), or the channel is specified to be invalid, the transfer is stopped. Every time a DMA transfer (task) that uses primary data or alternative data is complete, a transfer completion interrupt occurs.

Preparation:

Prepare primary data and alternative data, and set "1" to the bits of the channels corresponding to both DMAxCfg<master_enable> and DMAxChnlEnableSet.

Task A: Primary data
<cycle_ctrl[2:0]> = "011"
(ping-pong mode)
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x005" (6 times)



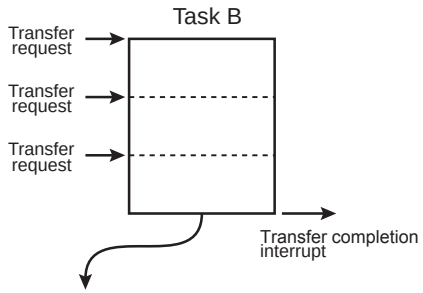
Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, the DMA performs remaining transfers twice toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task A, primary data for Task C can be set.

Task B: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x00B" (12 times)



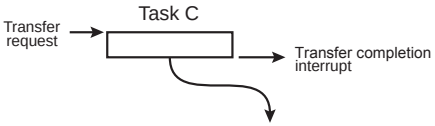
Receiving a transfer request, The DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, The DMA performs transfers twice toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task B, alternative data for Task D can be set.

Task C: Primary data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0001"
(2 times)
<n_minus_1[9:0]> = "0x001" (2 times)

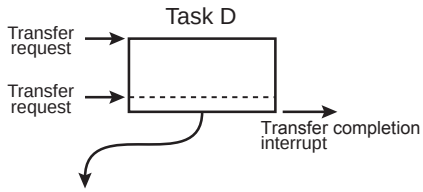


Receiving a transfer request, the DMA performs a transfer twice and performs arbitration.

The DMA generates a transfer completion interrupt request and performs an arbitration.

After completing Task C, alternative data for Task E can be set.

Task D: Alternative data
<cycle_ctrl[2:0]> = "011"
<R_power[4:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x004" (5 times)

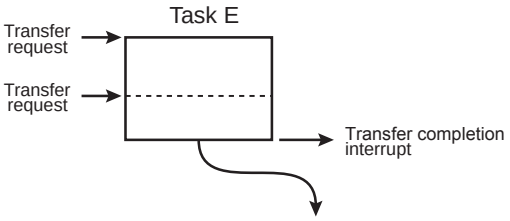


Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, the DMA performs a transfer once toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

Task E: Primary data
<cycle_ctrl[2:0]> = "011"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> = "0x006" (7 times)



Receiving a transfer request, the DMA performs a transfer four times and performs arbitration.

If there is no other high-priority requests, the DMA performs transfers three times toward a request for a transfer to the corresponding channels.

The DMA generates a transfer completion interrupt request and performs an arbitration.

Final: Alternative data
<cycle_ctrl[2:0]> = "000"
(invalid)

Even receiving a transfer request, the operation stops because <cycle_ctrl[2:0]> is set to invalid.

(The operation can be also stopped by setting the <cycle_ctrl[2:0]> of Task E to normal mode "001".)

10.4.3.5 Memory Scatter/Gather Mode

In memory scatter/gather mode, primary data is used in order to transfer data for alternative data.

Receiving a transfer request, the DMA transfers four alternative data using primary data. If there is no new requests, it starts data transferring using alternative data. Then, it keeps transferring alternative data using primary data and transfer using alternative data, until either invalid setting ("000") of the <cycle_ctrl [2:0]> or setting data of the basic mode ("001") is read. A new transfer request is not required during this period. After the transfer operation, an interrupt is generated.

The settings of the channel_cfg of primary data must be configured as shown below:

Table 10-3 Setting values of Memory scatter/gather mode (Primary data)

Bit	Bit symbol	Setting values	Description
[31:30]	dst_inc	10	4-byte increment is specified for transfer destination address.
[29:28]	dst_size	10	4 bytes are specified as transfer destination address.
[27:26]	src_inc	10	4-byte increment is specified for transfer source address.
[25:24]	src_size	10	4 bytes are specified as transfer source address.
[17:14]	R_power	0010	4 is specified as arbitration cycle.
[13:4]	n_minus_1	N	The number of alternative task to be prepared ×4 is specified.
[3]	next_useburst	0	"0" is specified in memory scatter/gather mode.
[2:0]	cycle_ctrl	100	Memory scatter/gather mode (primary data) is specified. note)

Note: If the transfers specified in the <n_minus_1> are complete, invalid data "000" is automatically set.

Preparation:

Prepare primary data. Set "100" to <cycle_ctrl> and set four task data 4 × 4 = 16 as the number of transfers <n_minus_1>.

Set alternative data for Task A,B,C and D to the memory location which is set to the <src_data_end_ptr>.

Set "1" to bits of channels corresponding to DMAxCfg <master_enable> and DMAxChnlSet.

Copy A: Primary data

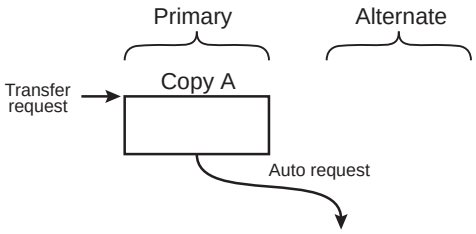
<cycle_ctrl[2:0]> = "100"

(Memory scatter / gather mode)

<R_power[3:0]> = "0010"

(4 times)

<n_minus_1[9:0]> = "0x00F" (16 times)



Receiving a transfer request, the DMA performs a transfer for alternative data of Task A for four times.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

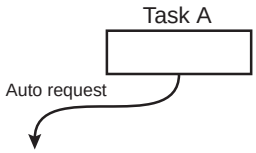
Task A: Alternative data

<cycle_ctrl[2:0]> = "100"

<R_power[3:0]> = "0010"

(4 times)

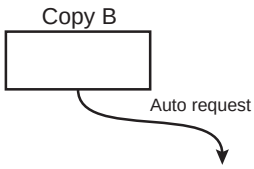
<n_minus_1[9:0]> = "0x002" (3 times)



The DMA performs Task A.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Copy B: Primary data



The DMA performs transfers for alternative data of Task B for four times.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

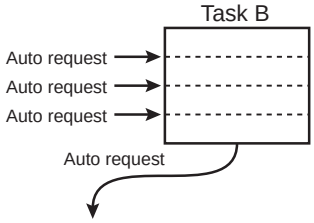
Task B: Alternative data

<cycle_ctrl[2:0]> = "100"

<R_power[3:0]> = "0001"

(2 times)

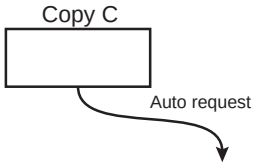
<n_minus_1[9:0]> = "0x007" (8 times)



The DMA performs Task B.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

Copy C: Primary data



The DMA performs transfers for alternative data of Task C for four times.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

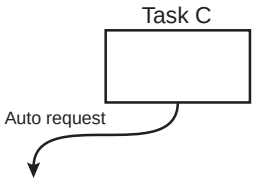
Task C: Alternative data

<cycle_ctrl[2:0> = "100"

<R_power[3:0]> = "0011"

(8 times)

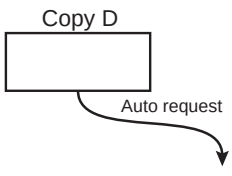
<n_minus_1[9:0]> = "0x004" (5 times)



The DMA performs Task C.

After completing the transfer, a transfer request is automatically generated and arbitration starts.

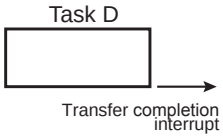
Copy D: Primary data



The DMA performs transfers for alternative data of Task D for four times. The DMA also sets "000" to <cycle_ctrl> of the primary data in order to set the next primary data invalid.

A transfer request is automatically generated and arbitration starts.

Task D: Alternative data
<cycle_ctr[2:0]l> = "001"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> =
"0x003" (4 times)



The DMA performs Task D.
Since <cycle_ctrl[2:0]> is set to the basic mode "001", the DMA generates a transfer completion interrupt request after the end of the transfer, and completes the operation.

10.4.3.6 Peripheral Scatter/Gather Mode

Primary data is used in order to transfer data for alternative data in peripheral scatter/gather mode.

Receiving a transfer request, the DMA transfers four alternative data using primary data, and then starts transfer using alternative data.

After that, if a transfer request is generated, it starts alternative data transferring using primary data. Then, it keeps transferring alternative data using primary data and transfer using alternative data, until either invalid setting ("000") of the <cycle_ctrl> or setting data of the basic mode ("001") is read. A new transfer request is not required during this period. After the transfer operation, an interrupt is generated.

The settings of the channel_cfg of primary data must be configured as shown below:

Table 10-4 Fixed values in peripheral scatter / gather mode (Primary data)

Bit	Bit symbol	Setting value	Description
[31:30]	dst_inc	10	A 4-byte increment is specified for transfer destination address.
[29:28]	dst_size	10	4 bytes are specified as transfer destination address.
[27:26]	src_inc	10	A 4-byte increment is specified for transfer source address.
[25:24]	src_size	10	4 bytes are specified as transfer source address.
[17:14]	R_power	0010	4 is specified as arbitration cycle.
[13:4]	n_minus_1	N	The number of alternative task to be prepared ×4 is specified.
[2:0]	cycle_ctrl	110	Specify peripheral scatter/gather mode (Primary data).

Note: If the transfers specified in the <n_minus_1> are complete, invalid data "000" is automatically set.

Preparation:

Prepare primary data. Set "110" to <cycle_ctrl> and 4 × 4 = 16 for four tasks to the number of transfers <n_minus_1>.

Set alternative data for Task A,B,C and D to the memory location which is set to the <src_data_end_ptr>.

Set "1" to bits of channels corresponding to DMAxCfg <master_enable> and DMAxChnlEnableSet.

Copy A: Primary data

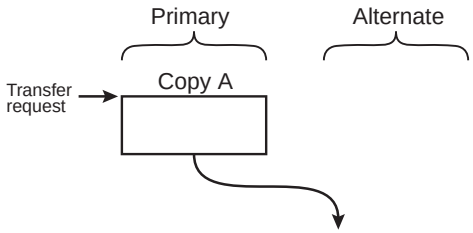
<cycle_ctrl[2:0]> = "110"

(Peripheral scatter / gather)

<R_power[3:0]> = "0010"

(4 times)

<n_minus_1[9:0]> = "0x00F" (16 times)



Receiving a transfer request, the DMA performs transfers for alternative data of Task A for four times.

After completing the transfer, operation automatically moves onto Task A.

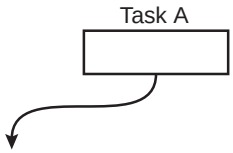
Task A: Alternative data

<cycle_ctrl[2:0]> = "111"

<R_power[3:0]> = "0010"

(4 times)

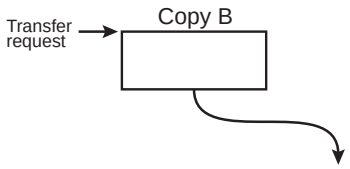
<n_minus_1[9:0]> = "0x002" (3 times)



The DMA performs Task A.

After completing the transfer, if a transfer request is sent from peripheral function and if it is high-priority request, the next operation starts.

Copy B: Primary data



The DMA performs transfers for alternative data of Task B for four times.

After completing the transfer, processing of Task B automatically starts.

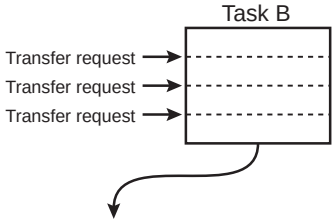
Task B: Alternative data

<cycle_ctrl[2:0]> = "111"

<R_power[3:0]> = "0001"

(2 times)

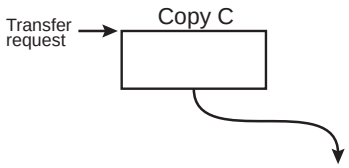
<n_minus_1[9:0]> = "0x007" (8 times)



The DMA performs Task B. Since an arbitration occurs every 2^R times of transfers, three times of transfer is required at least to complete Task B.

After completing the transfer, if a transfer request is sent from peripheral function and if it is high-priority request, the next operation starts

Copy C: Primary data



The DMA performs transfers for alternative data of Task C for four times.

After completing the transfer, operation automatically moves onto Task C.

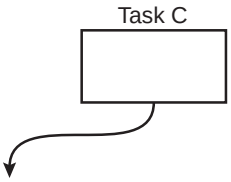
Task C: Alternative data

<cycle_ctrl[2:0]> = "111"

<R_power[3:0]> = "0011"

(8 times)

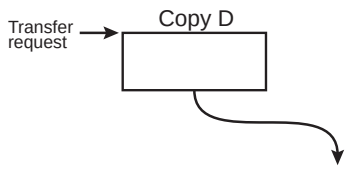
<n_minus_1[9:0]> = "0x004" (5 times)



The DMA performs Task C.

After completing the transfer, if a transfer request is sent from peripheral function and if it is high-priority request, the next operation starts.

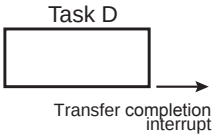
Copy D: Primary data



The DMA performs transfers for alternative data of Task D for four times. Also, The DMA performs transfers for alternative data for four times. Sets "000" to <cycle_ctrl> of the primary data and makes the next primary data invalid.

The operation automatically moves onto Task D.

Task D: Alternative data
<cycle_ctrl[2:0]> = "001"
<R_power[3:0]> = "0010"
(4 times)
<n_minus_1[9:0]> =
"0x003" (4 times)



The DMA performs Task D.
Since <cycle_ctrl> is set to the basic mode "001", The DMA generates a transfer completion interrupt request after the end of the transfer, and completes the operation.

10.5 Precautions

Extra caution should be exercised when a DMA transfer request is used in the following peripheral functions:

- Serial channel with 4-byte FIFO (SIO/UART)
- 16-bit timer/event counter (TMRB)
- Analog-to-digital converter (ADC)

10.5.1 SIO/UART, TMRB, ADC are Used

The following points should be considered:

- It is recommended to use the basic mode as a transfer mode.
- Set "after 1 transfer" as a DMA transfer rate.

  Specify "0000" as the arbitration rate <R_power> for the control data.
- Do not use the FIFO of the SIO/UART.

  Use the SIO/UART with the single-buffer or double-buffer setting.

A new request occurs after the DMA transfer is started, only one transfer is performed. Design the program to perform a DMA transfer surely.

In case that transfer will not be started, the following circumstances can be expected:

- A higher priority transfer request occurs in the same unit
- A transfer destination conflict occurs between other higher bus master and a sender.

As a guide, this µDMA controller takes 11 clocks on pre-/post-processing. It takes approximately 5 clocks for a data transfer between the peripheral functions and internal RAM.

11. Input / Output port

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

11.1 Registers

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

All registers are 32-bits. The configurations are different depend 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.

11.1.1 Register list

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

Register name	Address (Base+)	PORT A	PORT B	PORT C	PORT D	PORT E
Data register	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA
Output control register	0x0004	PACR	PBCR	PCCR	PDCR	PECR
Function register 1	0x0008	PAFR1	PBFR1	PCFR1	PDFR1	PEFR1
Function register 2	0x000C	PAFR2	-	-	PDFR2	PEFR2
Open-drain control register	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD
Pull-up control register	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP
Pull-down control register	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN
Input control register	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE

Register name	Address (Base+)	PORT F	PORT G	PORT H	PORT J	PORT K
Data register	0x0000	PFDATA	PGDATA	PHDATA	PJDATA	PKDATA
Output control register	0x0004	PFCR	PGCR	PHCR	PJCR	PKCR
Function register 1	0x0008	PFFR1	PGFR1	-	-	PKFR1
Function register 2	0x000C	PFFR2	-	-	-	-
Function register 3	0x0010	PFFR3	-	-	-	-
Open-drain control register	0x0028	PFOD	PGOD	PHOD	PJOD	PKOD
Pull-up control register	0x002C	PFPUP	PGPUP	PHPUP	PJPUP	PKPUP
Pull-down control register	0x0030	PFPDN	PGPDN	PHPDN	PJPDN	PKPDN
Input control register	0x0038	PFIE	PGIE	PHIE	PJIE	PKIE

Register name	Address (Base+)	PORT L	PORT N	PORT P
Data register	0x0000	PLDATA	PNDATA	PPDATA
Output control register	0x0004	PLCR	PNCR	PPCR
Function register 1	0x0008	PLFR1	PNFR1	-
Function register 2	0x000C	-	PNFR2	-
Open-drain control register	0x0028	PLOD	PNOD	PPOD
Pull-up control register	0x002C	PLPUP	PNPUP	PPPUP
Pull-down control register	0x0030	PLPDN	PNPDN	PPPDN
Input control register	0x0038	PLIE	PNIE	PPIE

Note: The address shown as "-" is not accessed.

11.1.2 Port function and setting list

The list of the function and setting register for each port is shown bellows.

- "Table 11-1 PORT A Setting List"
- "Table 11-2 PORT B Setting List"
- "Table 11-3 PORT C Setting List"
- "Table 11-4 PORT D Setting List"
- "Table 11-5 PORT E Setting List"
- "Table 11-6 PORT F Setting List"
- "Table 11-7 PORT G Setting List"
- "Table 11-8 PORT H Setting List"
- "Table 11-9 PORT J Setting List"
- "Table 11-10 PORT K Setting List"
- "Table 11-11 PORT L Setting List"
- "Table 11-12 PORT N Setting List"
- "Table 11-13 PORT P 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, PA0 represents that the port name A and bit 0 of its register.

11.1.2.1 PORT A

Table 11-1 PORT A Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	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
	INT3	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB0IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
PA1	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
	TB0OUT	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
PA2	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
	INT4	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB1IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
PA3	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
	TB1OUT	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
PA4	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
	SC1SCLK	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
		Output		0/1	1	PAFR1	0/1	0/1	0/1	0
	$\overline{\text{SC1CTS}}$	Input	FT2	0/1	0	PAFR2	0/1	0/1	0/1	1
PA5	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
	SC1TXD	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	TB6OUT	Output	FT1	0/1	1	PAFR2	0/1	0/1	0/1	0
PA6	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
	SC1RXD	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	TB6IN	Input	FT1	0/1	0	PAFR2	0/1	0/1	0/1	1
PA7	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
	INT8	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB4IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1

11.1.2.2 PORT B

Table 11-2 PORT B Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0	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
	TRACECLK0	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB1	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
	TRACEDATA0	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB2	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
	TRACEDATA1	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB3	After reset (TSM/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
	TMS	I/O	FT3	0/1	1	PBFR1	0/1	0/1	0/1	1
	SWDIO	I/O	FT3	0/1	1	PBFR1	0/1	0/1	0/1	1
PB4	After reset (TCK/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
	TCK	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
	SWCLK	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
PB5	After reset (TDO/SWV)			0	1	PBFR1	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
	TDO	Output	FT3	0/1	1	PBFR1	0/1	0/1	0/1	0
	SWV	Output	FT3	0/1	1	PBFR1	0/1	0/1	0/1	0
PB6	After reset (TDI)			0	0	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
	TDI	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
PB7	After reset ($\overline{\text{TRST}}$)			0	0	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
	$\overline{\text{TRST}}$	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1

11.1.2.3 PORT C

Table 11-3 PORT C Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE
PC0	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
	UO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC1	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
	XO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC2	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
	VO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC3	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
	YO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC4	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
	WO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC5	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
	ZO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC6	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
	$\overline{\text{EMG0}}$	Input	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1
PC7	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
	$\overline{\text{OVV0}}$	Input	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1

11.1.2.4 PORT D

Table 11-4 PORT D Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE
PD0	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
	ENCA0	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5IN	Input	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
PD1	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
	ENCB0	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5OUT	Output	FT1	0/1	1	PDFR2	0/1	0/1	0/1	0
PD2	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
	ENCZ0	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
PD3	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
	INT9	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
PD4	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
	SC2SCLK	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
		Output		0/1	1	PDFR1	0/1	0/1	0/1	0
	$\overline{\text{SC2CTS}}$	Input	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
PD5	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
	SC2TXD	Output	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
PD6	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
	SC2RXD	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1

11.1.2.5 PORT E

Table 11-5 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
	TB8OUT	Output	FT1	0/1	1	PEFR2	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
	TB8IN	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
PE3	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
	TB4OUT	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE4	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
	INT5	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB2IN	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE5	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
	TB2OUT	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0
PE6	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
	INT6	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB3IN	Input	FT1	0/1	0	PEFR1	0/1	0/1	0/1	1
PE7	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
	INT7	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB3OUT	Output	FT1	0/1	1	PEFR1	0/1	0/1	0/1	0

11.1.2.6 PORT F

Table 11-6 PORT F Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				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
PF1	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
	TB7OUT	Output	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF2	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
	ENCA1	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3SCLK	Input	FT1	0/1	0	PFFR2	0/1	0/1	0/1	1
		Output		0/1	1	PFFR2	0/1	0/1	0/1	0
	$\overline{\text{SC3CTS}}$	Input	FT1	0/1	0	PFFR3	0/1	0/1	0/1	1
PF3	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
	ENCB1	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3TXD	Output	FT1	0/1	1	PFFR2	0/1	0/1	0/1	0
PF4	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
	ENCZ1	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3RXD	Input	FT1	0/1	0	PFFR2	0/1	0/1	0/1	1

Note: 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.

11.1.2.7 PORT G

Table 11-7 PORT G Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PG0	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
	UO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG1	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
	XO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG2	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
	VO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG3	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
	YO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG4	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
	WO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG5	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
	ZO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG6	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
	$\overline{\text{EMG1}}$	Input	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
PG7	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
	$\overline{\text{OVV1}}$	Input	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1

11.1.2.8 PORT H

Table 11-8 PORT H Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PHDATA	PHCR	PHFRn	PHOD	PHPUP	PHPDN	PHIE
PH0	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA0	Input	FT5	0/1	0		0/1	0	0	0
	INT0	Input	FT4	0/1	0		0/1	0	0/1	1
PH1	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA1	Input	FT5	0/1	0		0/1	0	0	0
	INT1	Input	FT4	0/1	0		0/1	0	0/1	1
PH2	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA2	Input	FT5	0/1	0		0/1	0	0	0
	INT2	Input	FT4	0/1	0		0/1	0	0/1	1
PH3	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA3	Input	FT5	0/1	0		0/1	0	0	0
PH4	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA4	Input	FT5	0/1	0		0/1	0	0	0
PH5	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA5	Input	FT5	0/1	0		0/1	0	0	0
PH6	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA6	Input	FT5	0/1	0		0/1	0	0	0
PH7	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINA7	Input	FT5	0/1	0		0/1	0	0	0

11.1.2.9 PORT J

Table 11-9 PORT J Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ0	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB3	Input	FT5	0/1	0		0/1	0	0	0
PJ1	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB4	Input	FT5	0/1	0		0/1	0	0	0
PJ2	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB5	Input	FT5	0/1	0		0/1	0	0	0
PJ3	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB6	Input	FT5	0/1	0		0/1	0	0	0
PJ4	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB7	Input	FT5	0/1	0		0/1	0	0	0
PJ5	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	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	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB9	Input	FT5	0/1	0		0/1	0	0	0
	INTC	Input	FT4	0/1	0		0/1	0/1	0/1	1
PJ7	After reset			0	0		0	0	0	0
	Input Port	Input		0	0		0/1	0/1	0/1	1
	Output Port	Output		0	1		0/1	0/1	0/1	0
	AINB10	Input	FT5	0/1	0		0/1	0	0	0
	INTD	Input	FT4	0/1	0		0/1	0/1	0/1	1

11.1.2.10 PORT K

Table 11-10 PORT K Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PKDATA	PKCR	PKFRn	PKOD	PKPUP	PKPDN	PKIE
PK0	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
	INTE	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TRACEDATA3	Output	FT1	0/1	1	PKFR1	0/1	0/1	0/1	0
PK1	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
	INTF	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TRACEDATA2	Output	FT1	0/1	1	PKFR1	0/1	0/1	0/1	0

11.1.2.11 PORT L

Table 11-11 PORT L Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE
PL0	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
	INTB	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	CA_TX	Output	FT1	0/1	1	PLFR1	0/1	0/1	0/1	0
PL1	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
	INTA	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	CA_RX	Input	FT1	0/1	0	PLFR1	0/1	0/1	0/1	1

11.1.2.12 PORT N

Table 11-12 PORT N Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PNDATA	PNCR	PNFRn	PNOD	PNPUP	PNPDN	PNIE
PN0	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
	SB0SDA	I/O	FT1	0/1	1	PNFR2	1	0/1	0/1	1
PN1	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
	TB9OUT	Output		0/1	1	PNFR1	0/1	0/1	0/1	0
	SB0SCL	I/O	FT1	0/1	1	PNFR2	1	0/1	0/1	1
PN2	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
	TB9IN	Input	FT1	0/1	0	PNFR1	0/1	0/1	0/1	1

11.1.2.13 PORT P

Table 11-13 PORT P Setting List

PORT	Reset status	Input/Output	PORT Type	Control registers						
				PPDATA	PPCR	PPFRn	PPOD	PPPUP	PPPDN	PPIE
PP0	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
	AINA8	Input	FT5	0/1	0		0/1	0	0	0
PP1	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
	AINA9	Input	FT5	0/1	0		0/1	0	0	0
	AINB0	Input	FT5	0/1	0		0/1	0	0	0
PP2	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
	AINA10	Input	FT5	0/1	0		0/1	0	0	0
	AINB1	Input	FT5	0/1	0		0/1	0	0	0
PP3	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
	AINA11	Input	FT5	0/1	0		0/1	0	0	0
	AINB2	Input	FT5	0/1	0		0/1	0	0	0

11.2 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.

11.2.1 Type FT1

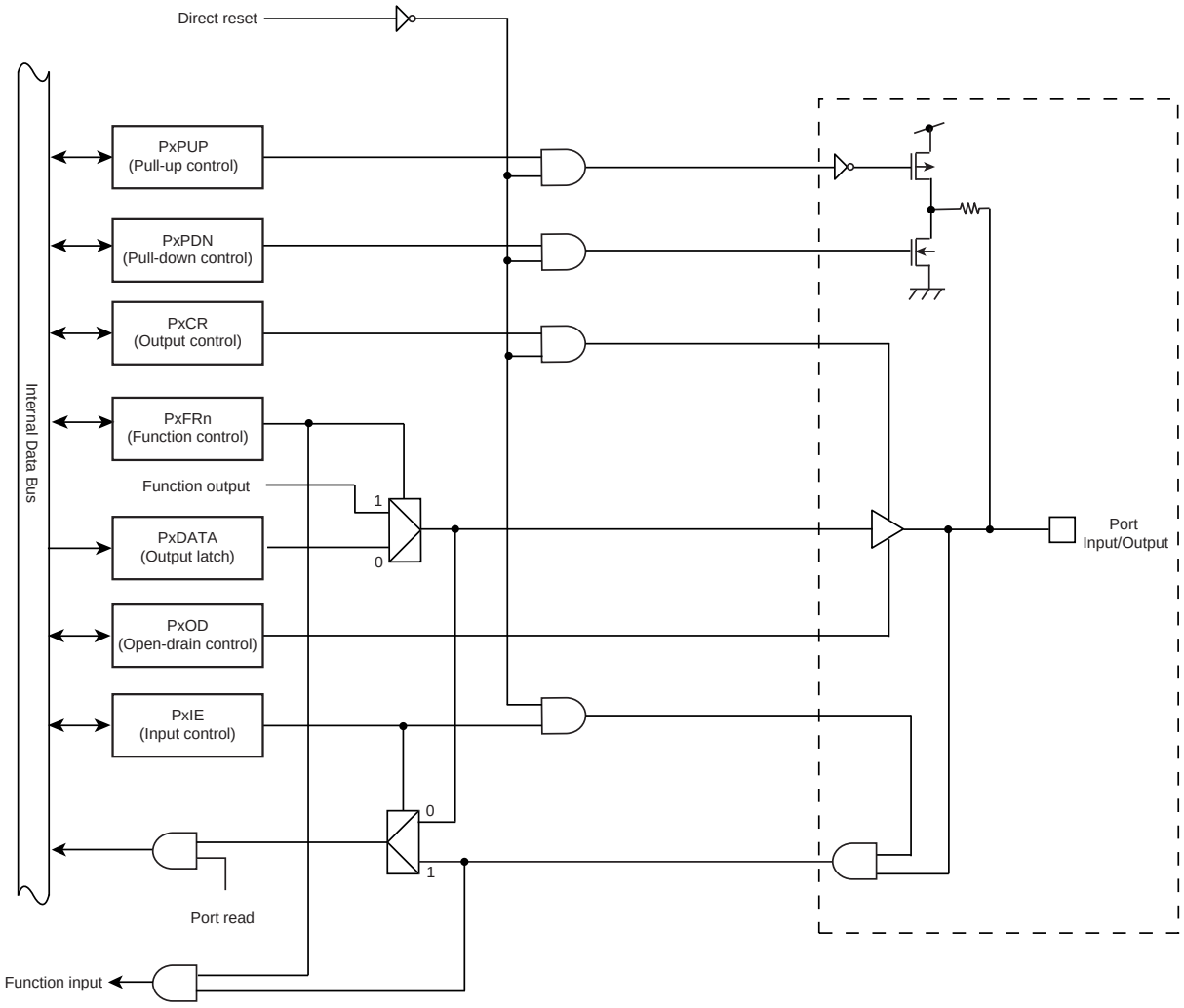


Figure 11-1 Port Type FT1

11.2.2 Type FT2

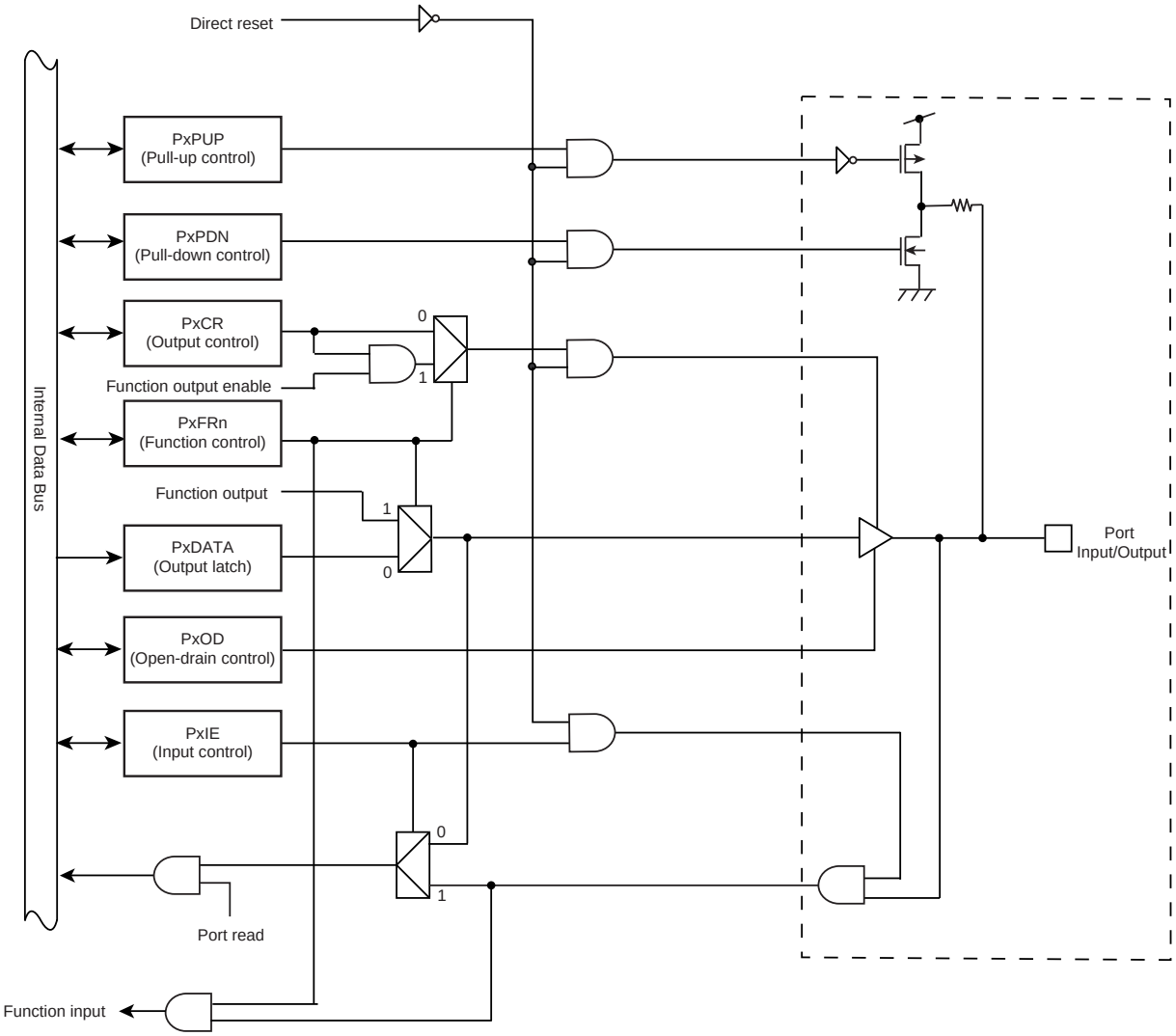


Figure 11-2 Port Type FT2

11.2.3 Type FT3

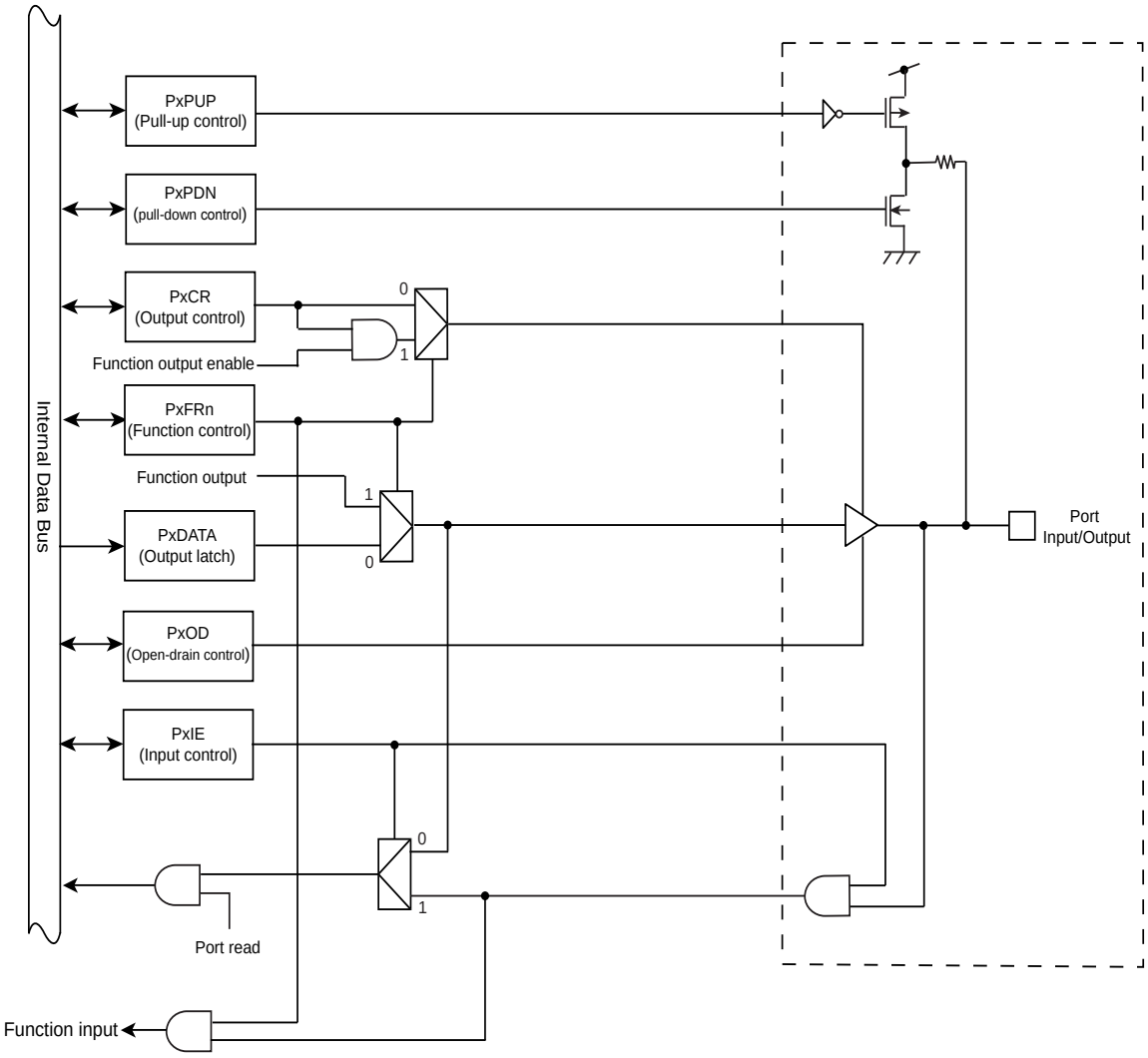


Figure 11-3 Port Type FT3

11.2.4 Type FT4

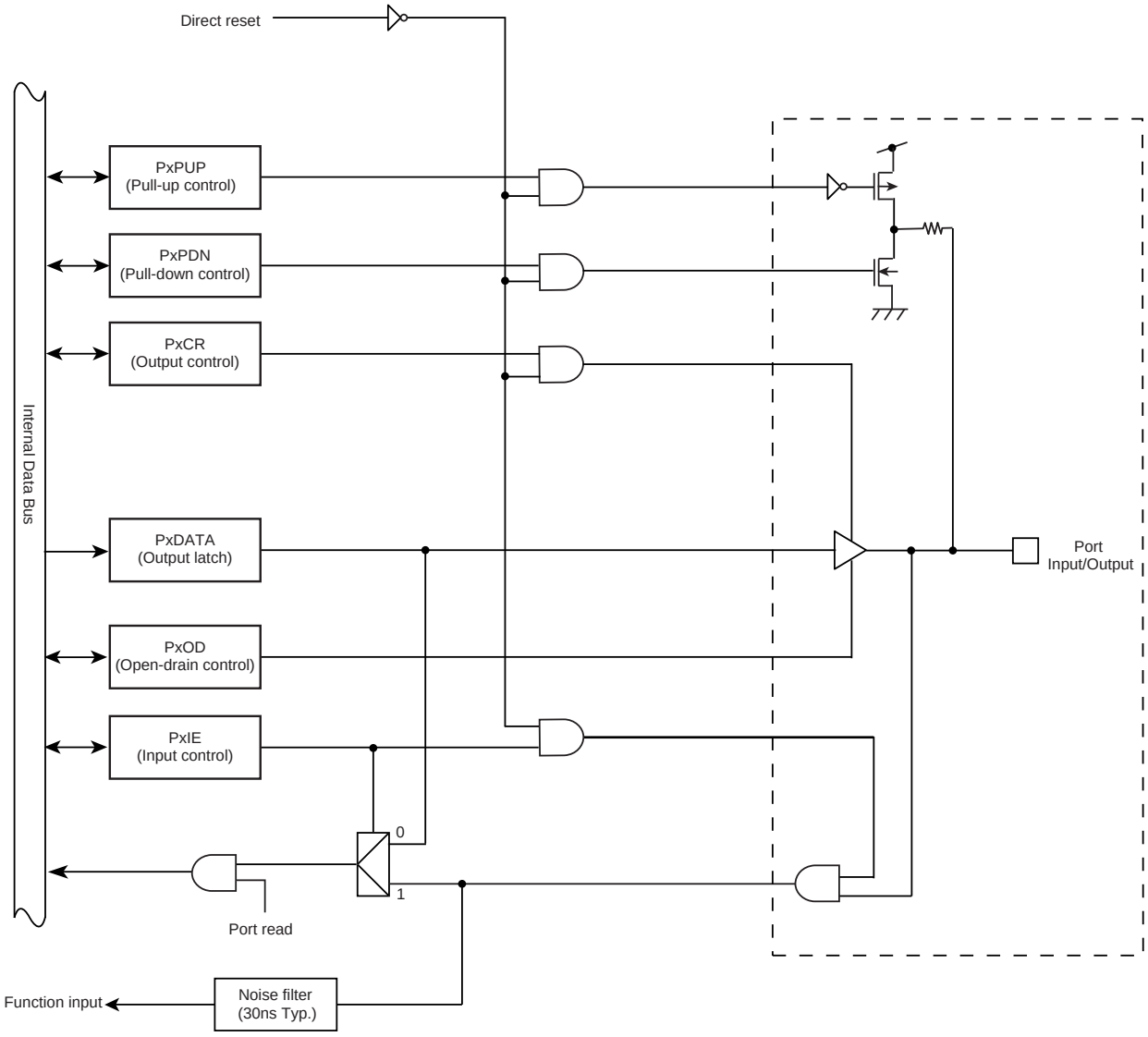


Figure 11-4 Port Type FT4

11.2.5 Type FT5

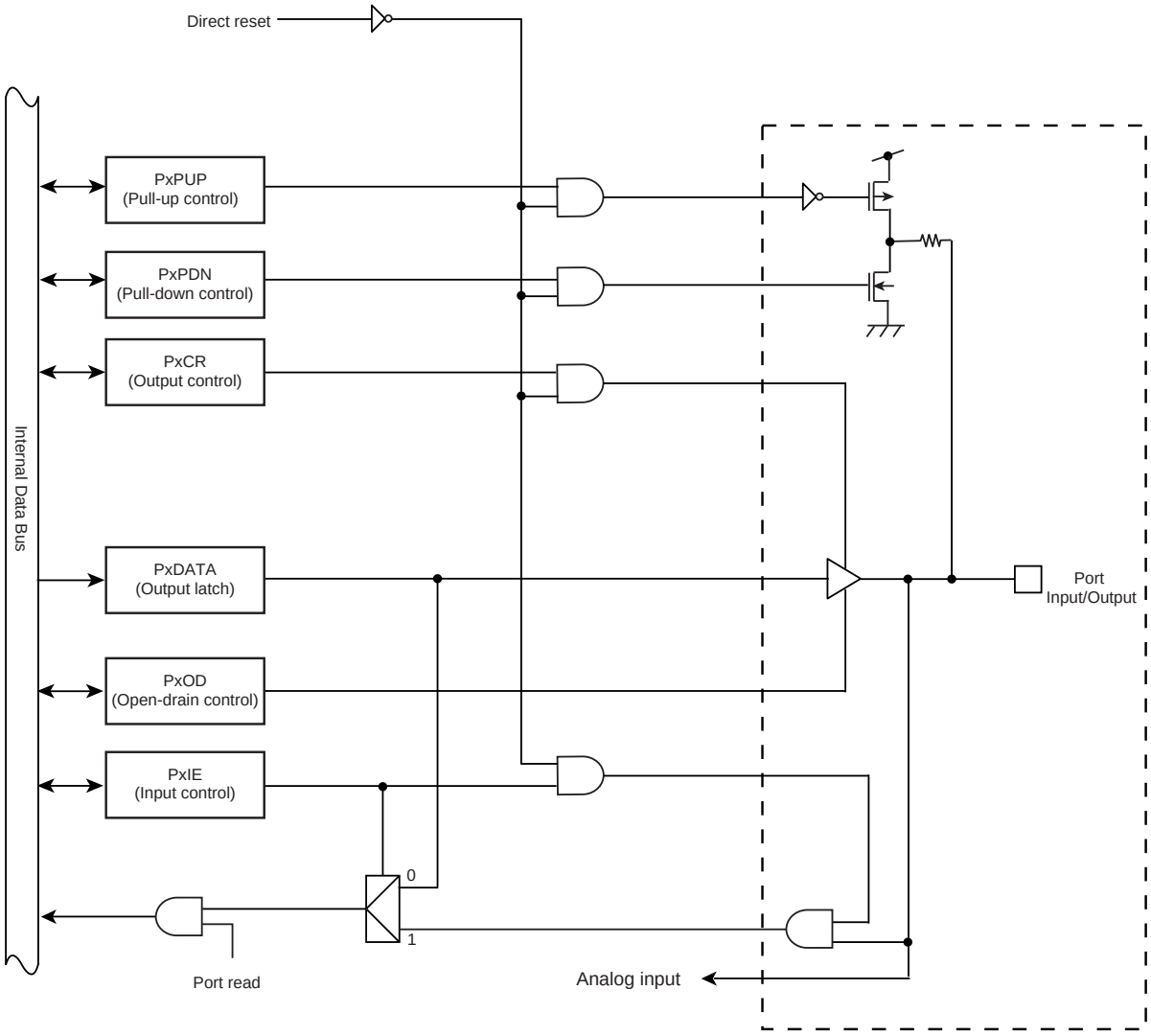


Figure 11-5 Port Type FT5

11.2.6 Type FT6

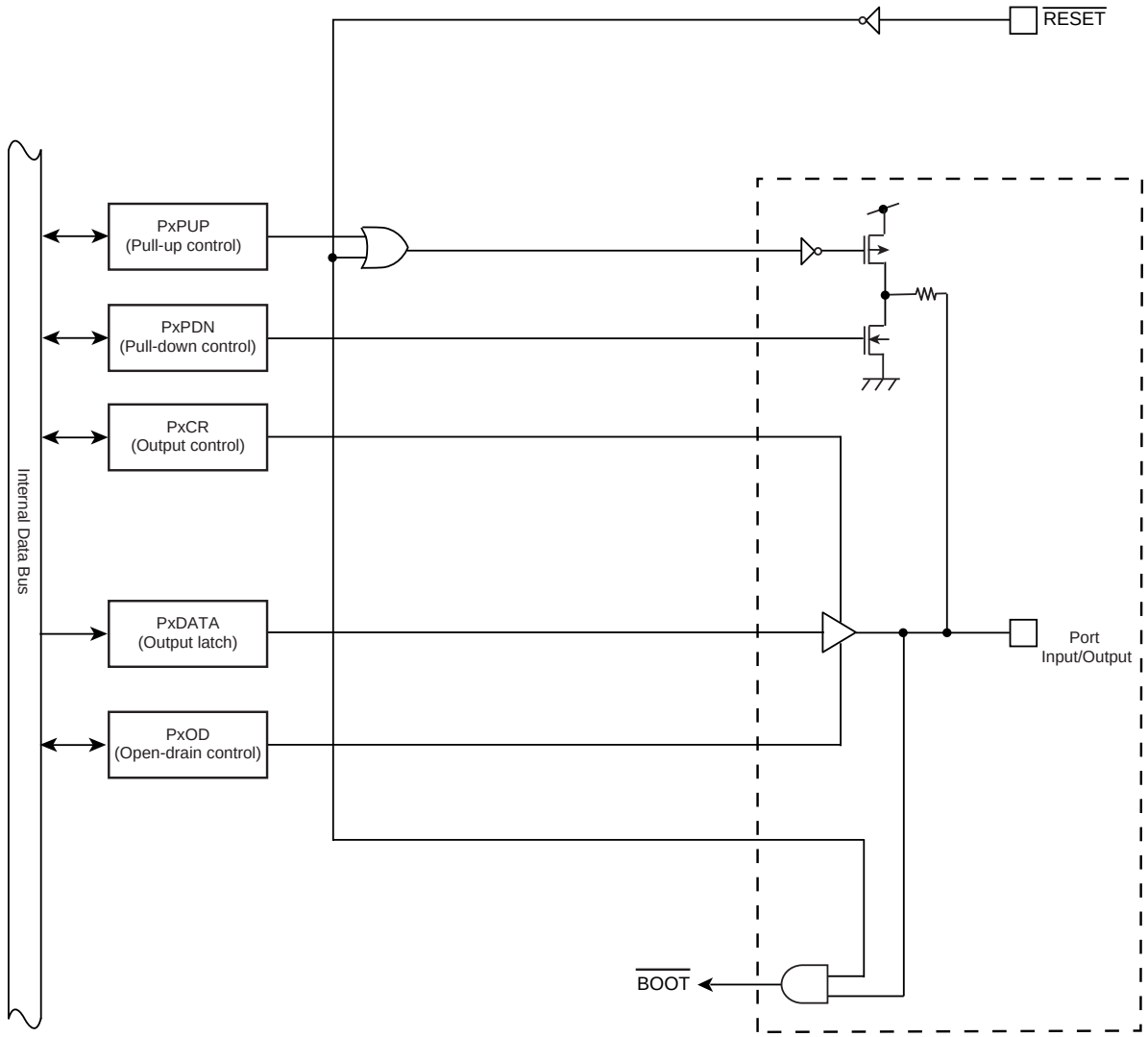


Figure 11-6 Port Type FT6

11.2.7 Type FT7

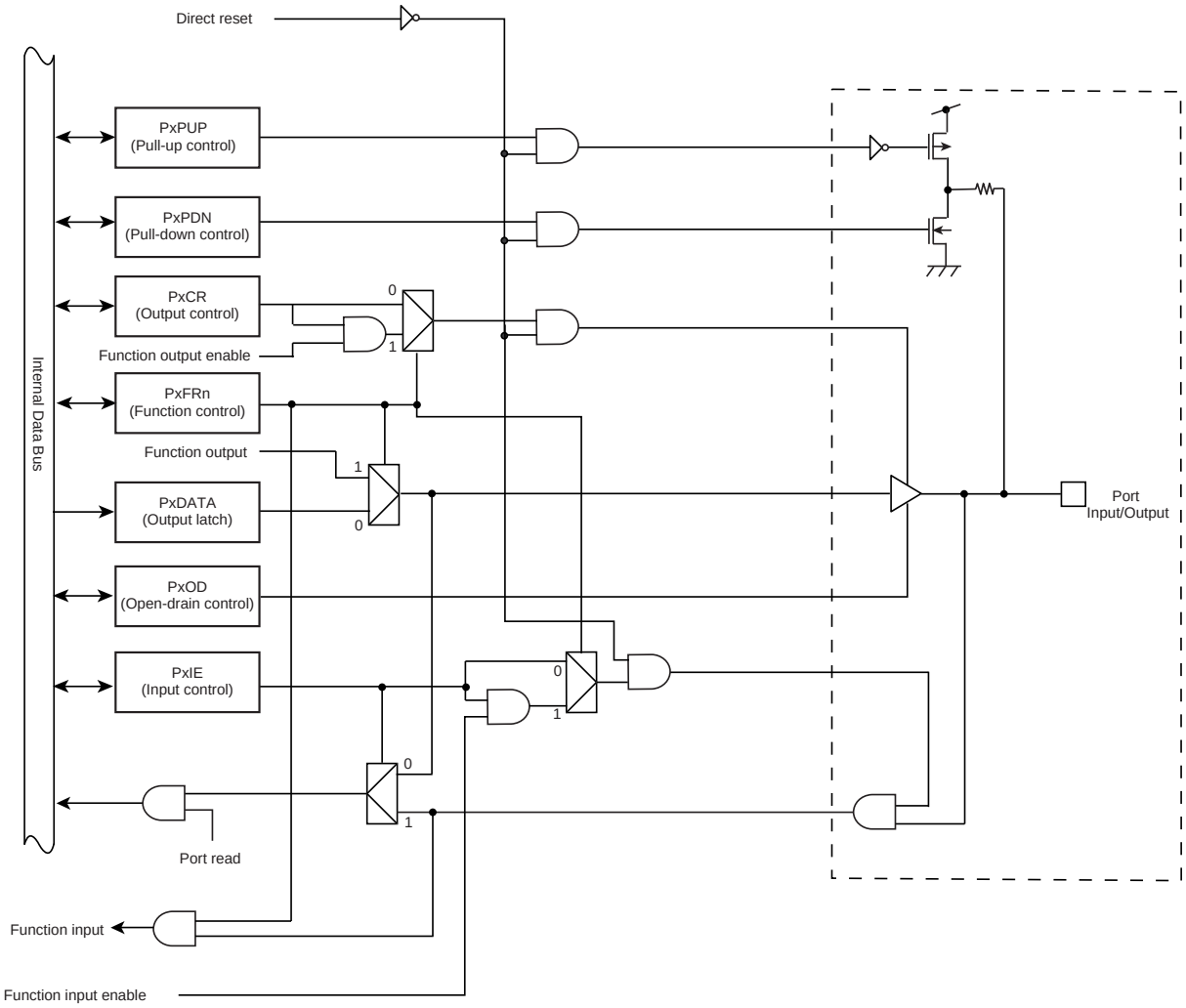


Figure 11-7 Port Type FT7

11.2.8 Type FT8

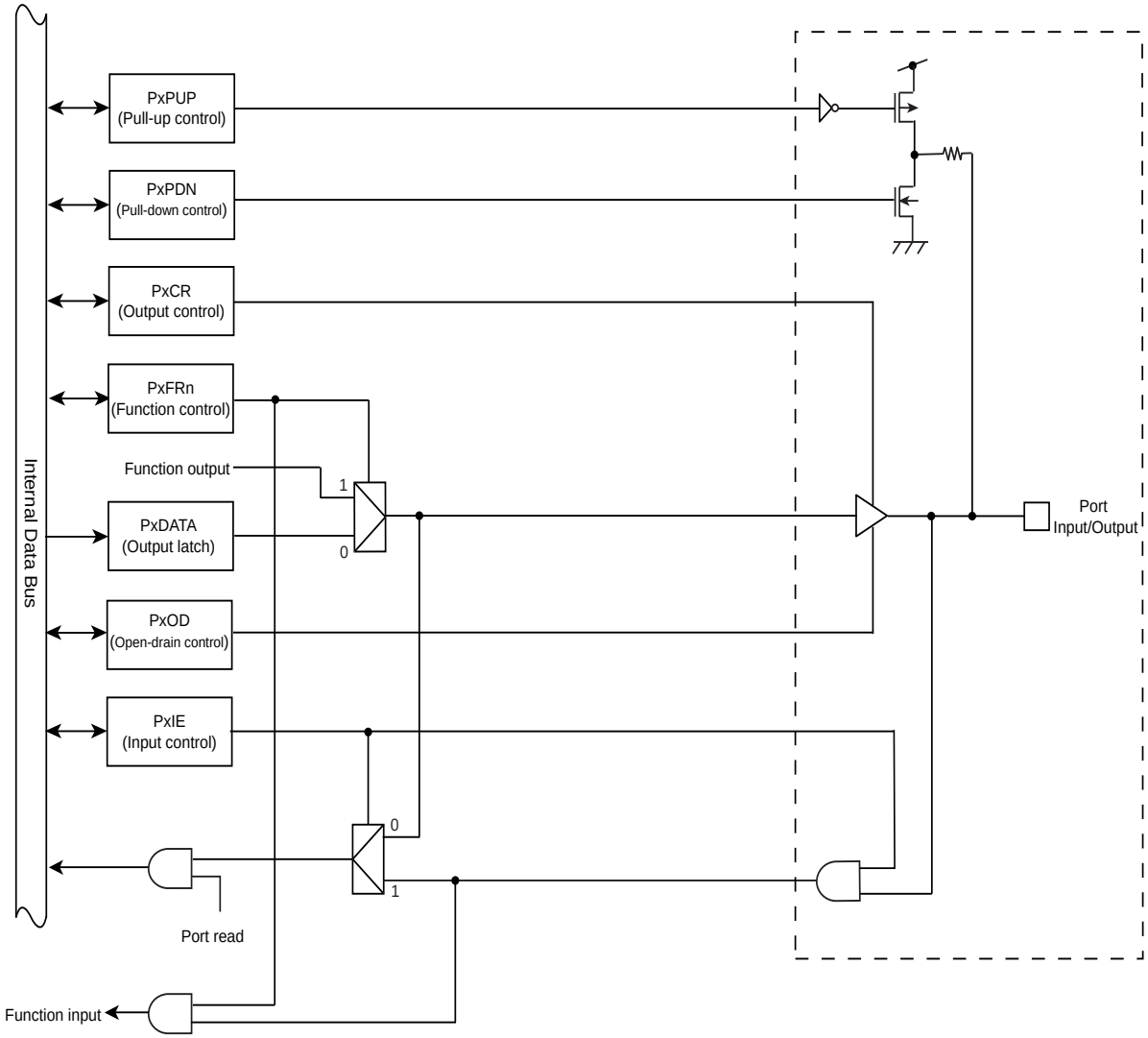


Figure 11-8 Port Type FT8

12. 16-bit Timer / Event Counters (TMRB)

12.1 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.

12.2 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.

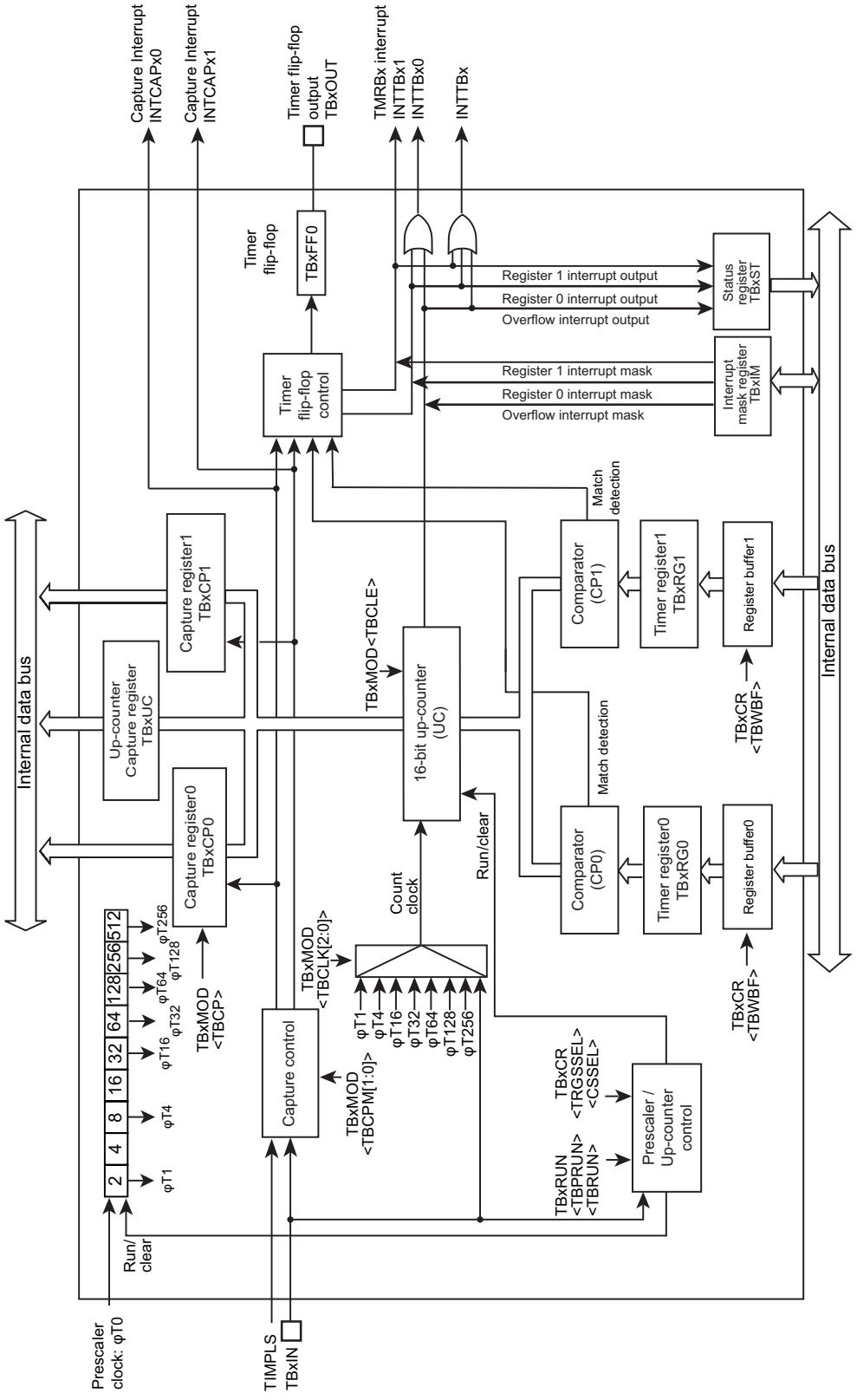


Figure 12-1 TMRBx Block Diagram

12.3 Registers

12.3.1 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

12.3.2 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".

12.3.3 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.

12.3.4 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.

12.3.5 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.

12.3.6 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.

12.3.7 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.

12.3.8 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.

12.3.9 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.

12.3.10 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.

12.3.11 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.

12.3.12 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.

12.3.13 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.

12.4 Description of Operations for Each Circuit

12.4.1 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.

12.4.2 Up-counter (UC)

UC is a 16-bit binary counter.

12.4.2.1 Source clock

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

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

12.4.2.2 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.

12.4.2.3 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".

12.4.2.4 Up-Counter Overflow

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

12.4.3 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.

12.4.4 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>.

12.4.5 Capture register (TBxCP0, TBxCP1)

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

12.4.6 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.

12.4.7 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.

12.4.8 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.

12.4.9 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.

12.5 Description of Operations for Each Mode

12.5.1 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

12.5.2 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

12.5.3 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

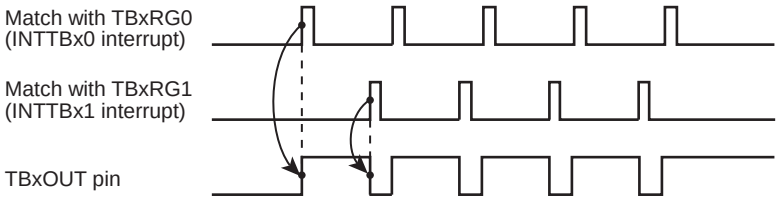


Figure 12-2 Example of Output of Programmable Pulse Generation (PPG)

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.

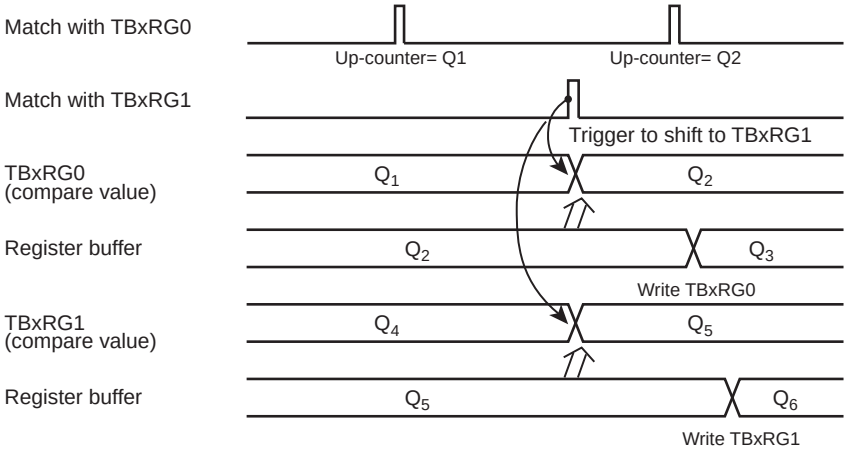


Figure 12-3 Register Buffer Operation

The block diagram of this mode is shown below.

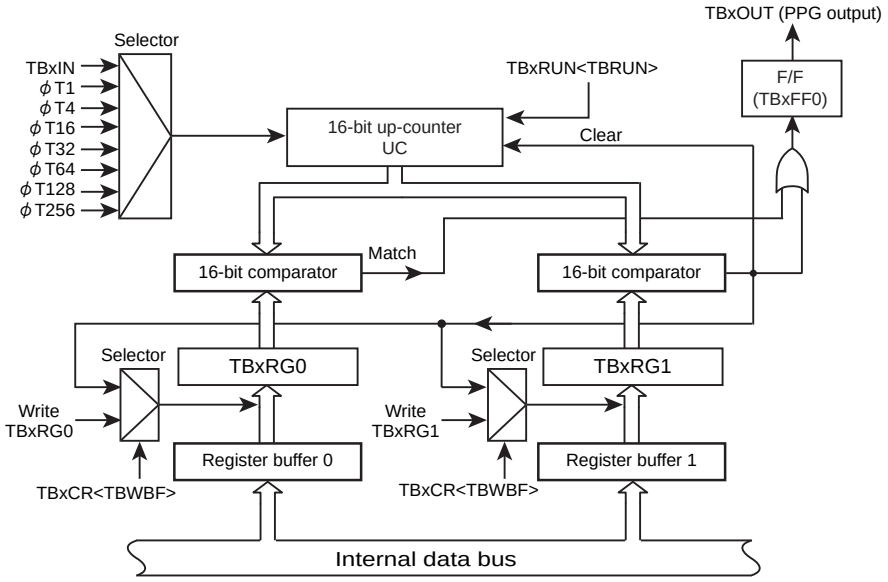


Figure 12-4 Block Diagram of 16-bit PPG Mode

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

12.5.4 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 12-5.

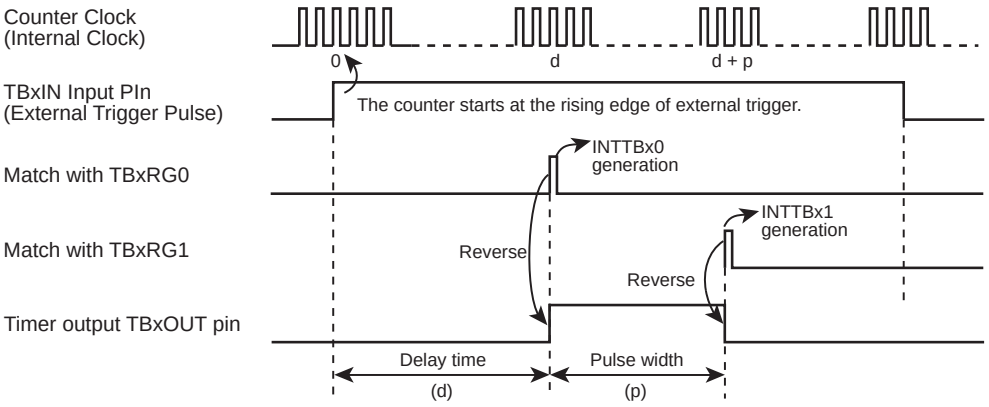


Figure 12-5 One-shot pulse generation using an external count start trigger (with a delay)

12.6 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

12.6.1 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 12-6.

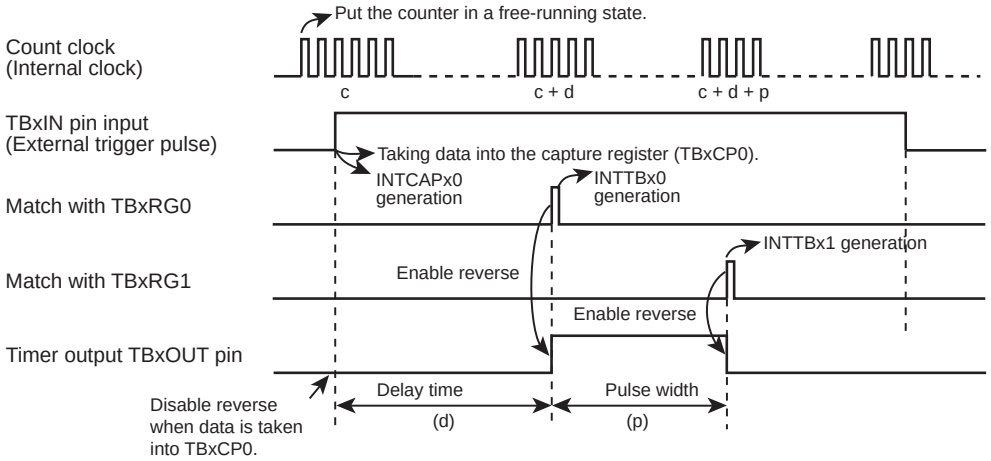


Figure 12-6 One-shot Pulse Output (With Delay)

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.

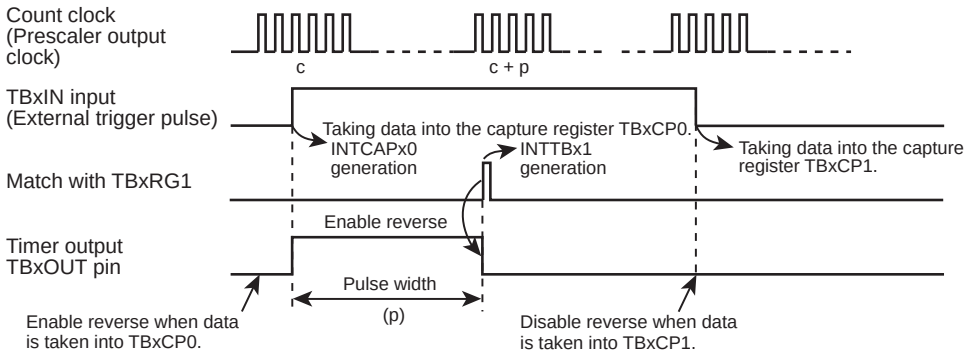


Figure 12-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

12.6.2 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 12-8 and this difference is multiplied by the cycle of the prescaler output clock to obtain the "Low" level width.

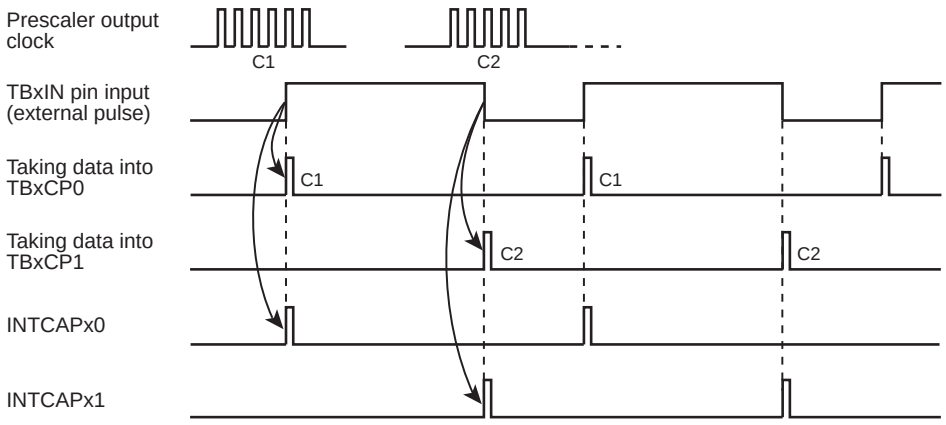


Figure 12-8 Pulse Width Measurement

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

13.1 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.

13.2 Configuration

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

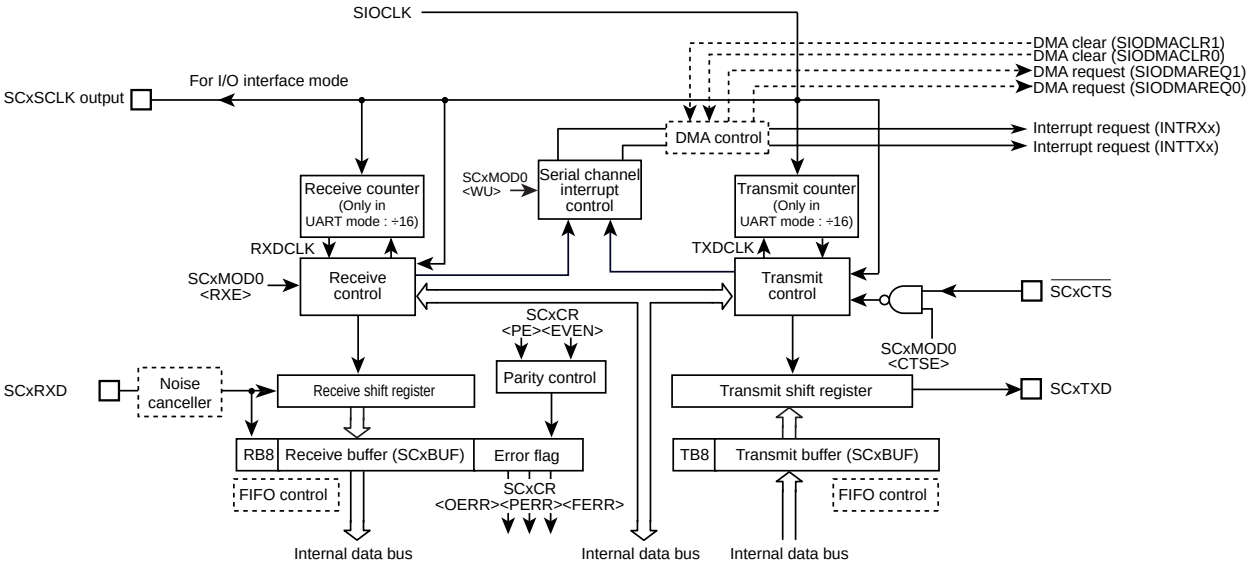


Figure 13-1 Serial Channel Block Diagram

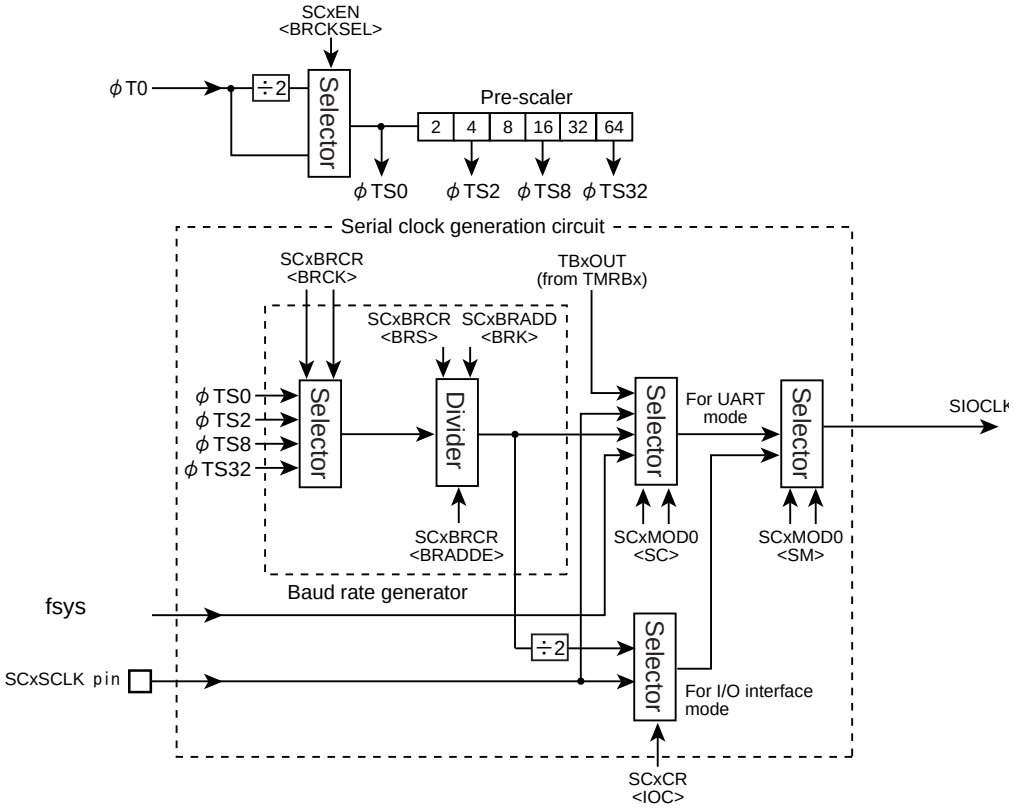


Figure 13-2 Serial clock generation circuit block diagram

13.3 Registers Description

13.3.1 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.

13.3.2 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.

13.3.3 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

13.3.4 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 100: 32/fsys 001: 4/fsys 101: 64/fsys 010: 8/fsys 110: 128/fsys 011: 16/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.

13.3.5 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.

13.3.6 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.

13.3.7 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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	TBEMP	R	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".
6	RBFLL	R	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 re-ceive double buffers, this bit changes to "1". When reading the receive buffer, this bit is cleared to "0".
5	TXRUN	R	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.
4	SBLEN	R/W	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>.
3	DRCHG	R/W	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.
2	WBUF	R/W	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>.

<TXRUN>	<TBEMP>	Status
1	−	Transmission in progress
0	1	Transmission is completed.
	0	Wait state with data in transmit buffer

<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.

13.3.8 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.

13.3.9 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 13-1 lists the settings of baud rate generator division ratio.

Table 13-1 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.

13.3.10 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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7-5	-	R/W	Be sure to write "000".
4	RFST	R/W	Bytes used in 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]>.
3	TFIE	R/W	Specify transmit interrupt for transmit FIFO. 0: Disabled 1: Enabled When transmit FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.
2	RFIE	R/W	Specify receive interrupt for receive FIFO. 0: Disabled 1: Enabled When receive FIFO is enabled, receive interrupts are enabled or disabled by this parameter.
1	RXTXCNT	R/W	Automatic disable of RXE/TXE. 0: None 1: Auto disable Controls automatic disabling of transmission and reception. Setting "1" enables to operate as follows. Half duplex Receive: 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. Half duplex Transmit: When the transmit shift register, transmit buffers and the transmit FIFO are empty, SCxMOD1<TXE> is automatically set to "0" to inhibit further transmission. Full duplex: When either of the above two conditions is satisfied, <TXE> and <RXE> are automatically set to "0" to inhibit further transmission and reception.
0	CNFG	R/W	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: Half duplex Receive: Receive FIFO 4bytes Half duplex Transmit: Transmit FIFO 4bytes Full duplex: Receive FIFO 2bytes and Transmit FIFO 2bytes

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.

13.3.11 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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	RFCS	W	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".
6	RFIS	R/W	Select interrupt generation condition. 0: When FIFO fill level (SCxRST<RLVL[2:0]>) = Receive FIFO fill level to generate receive interrupt (<RIL[1:0]>) 1: When FIFO fill level (SCxRST<RLVL[2:0]>) ≥ Receive FIFO fill level to generate receive interrupt (<RIL[1:0]>) For the detail of interrupt condition, refer to "13.13.1.2 FIFO"
5-2	-	R	Read as "0".
1-0	RIL[1:0]	R/W	FIFO fill level to generate receive interrupts.

	Half duplex	Full duplex
00	4 bytes	2 bytes
01	1 byte	1 byte
10	2 bytes	2 bytes
11	3 bytes	1 byte

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")

13.3.12 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 "13.13.2.2 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.

13.3.13 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.**

13.3.14 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.**

13.4 Operation in Each Mode

Table 13-2 shows the modes.

Table 13-2 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.

13.5 Data Format

13.5.1 Data Format List

Figure 13-3 shows data format.

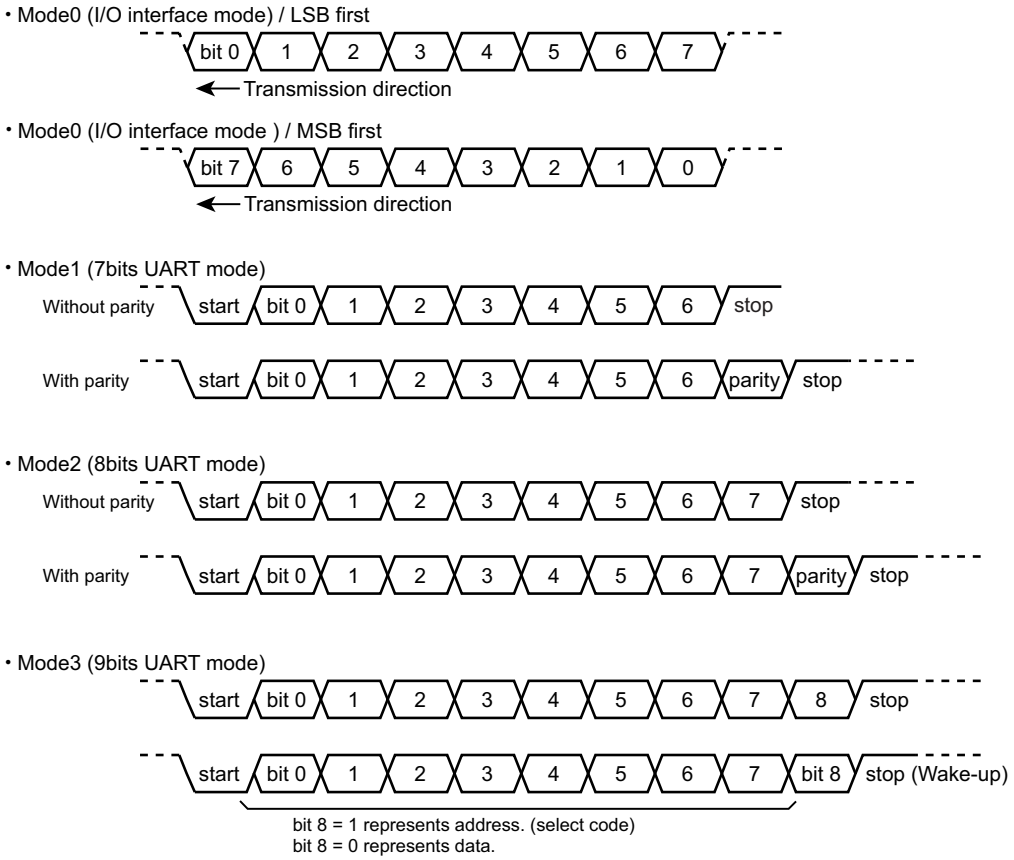


Figure 13-3 Data Format

13.5.2 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.

13.5.2.1 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.

13.5.2.2 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.

13.5.3 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.

13.6 Clock Control

13.6.1 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".

13.6.2 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.

13.6.2.1 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

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

13.6.2.2 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 13-3 shows clock selection in I/O interface mode.

Table 13-3 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 13-4 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 13-4 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

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR<PRCK[1:0]>}}{(\text{TBxRG1} \times 2) \times 2 \times 16}$$

In the case the timer prescaler clock ΦT1 (2division ratio) is selected.

One clock cycle is a period that the timer flip-flop is inverted twice.

13.7 Transmit/Receive Buffer and FIFO

13.7.1 Configuration

Figure 13-4 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.

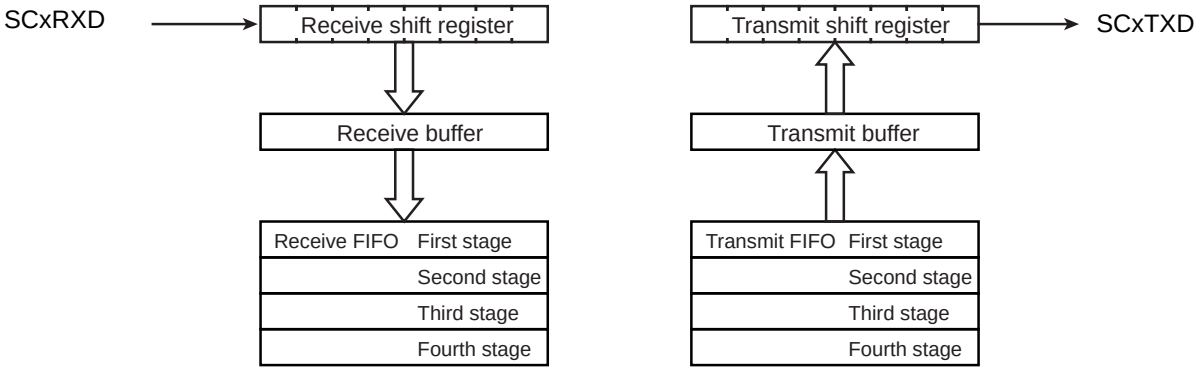


Figure 13-4 The Configuration of Buffer and FIFO

13.7.2 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 13-5 shows correlation between modes and buffers.

Table 13-5 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

13.7.3 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.

13.7.4 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 13-6 shows correction between modes and FIFO.

Table 13-6 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

13.8 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".

13.9 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

13.9.1 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 over-run 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.**

13.9.2 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.

13.9.3 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.

13.10 Receive

13.10.1 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.

13.10.2 Receive Control Unit

13.10.2.1 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>.

13.10.2.2 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.

13.10.3 Receive Operation

13.10.3.1 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.

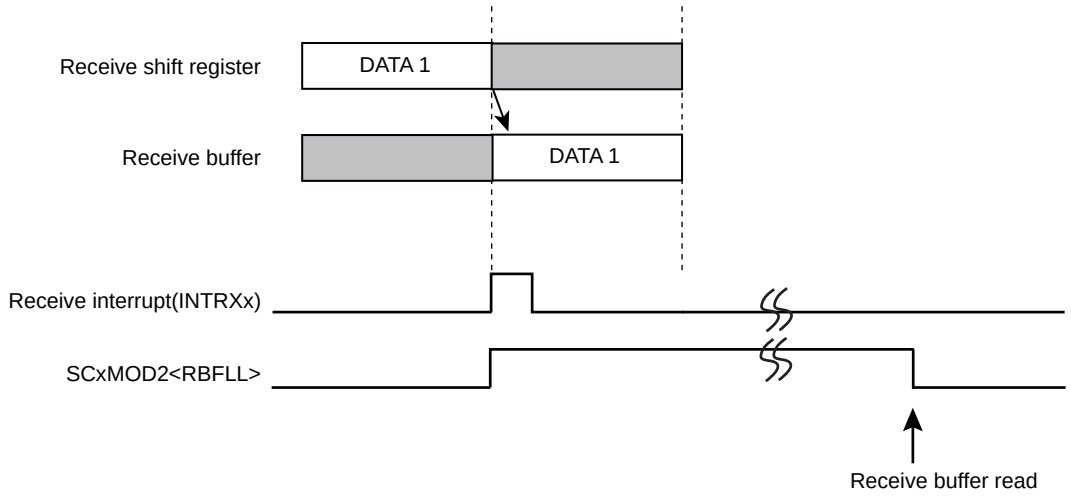


Figure 13-5 Receive Buffer Operation

13.10.3.2 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.

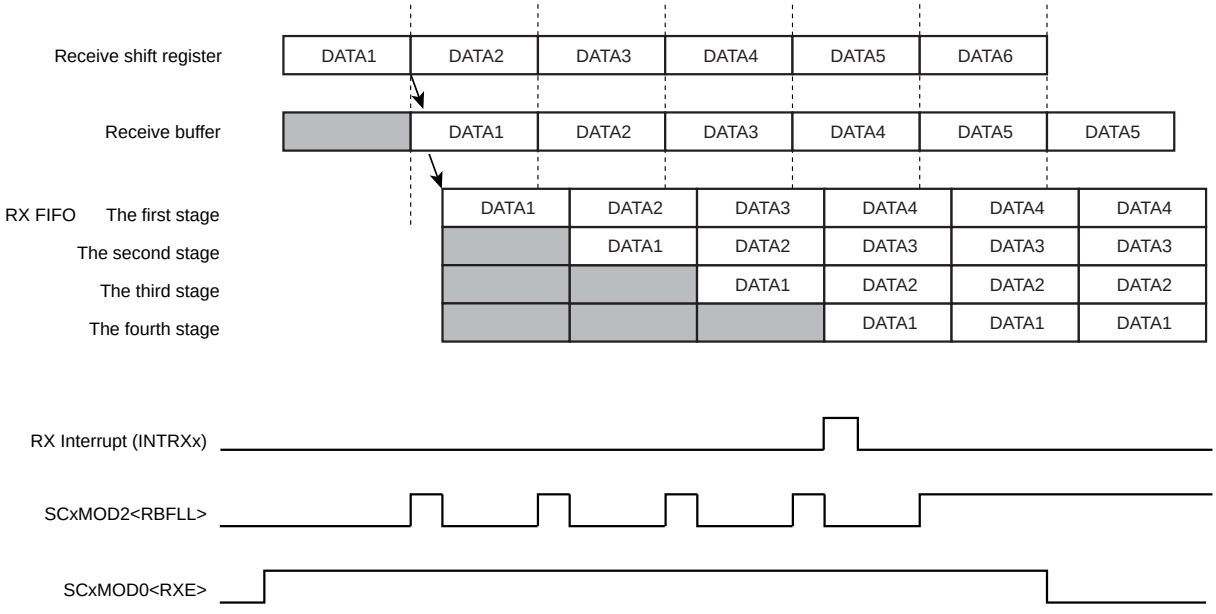


Figure 13-6 Receive FIFO Operation

13.10.3.3 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>.

13.10.3.4 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>.

13.10.3.5 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".

13.10.3.6 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.

13.11 Transmit

13.11.1 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.

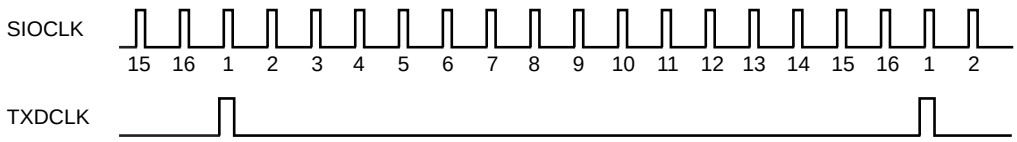


Figure 13-7 Generation of Transmission Clock in UART mode

13.11.2 Transmit Control

13.11.2.1 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>.

13.11.2.2 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.

13.11.3 Transmit Operation

13.11.3.1 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".

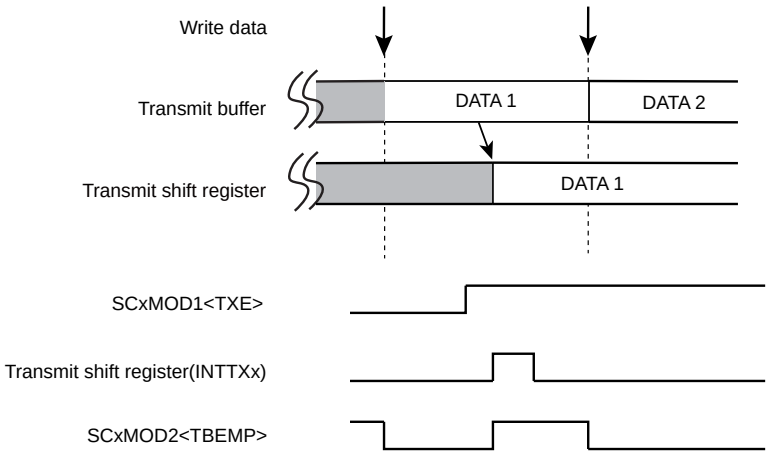


Figure 13-8 Operation of Transmit Buffer (Double-buffer is enabled)

13.11.3.2 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".

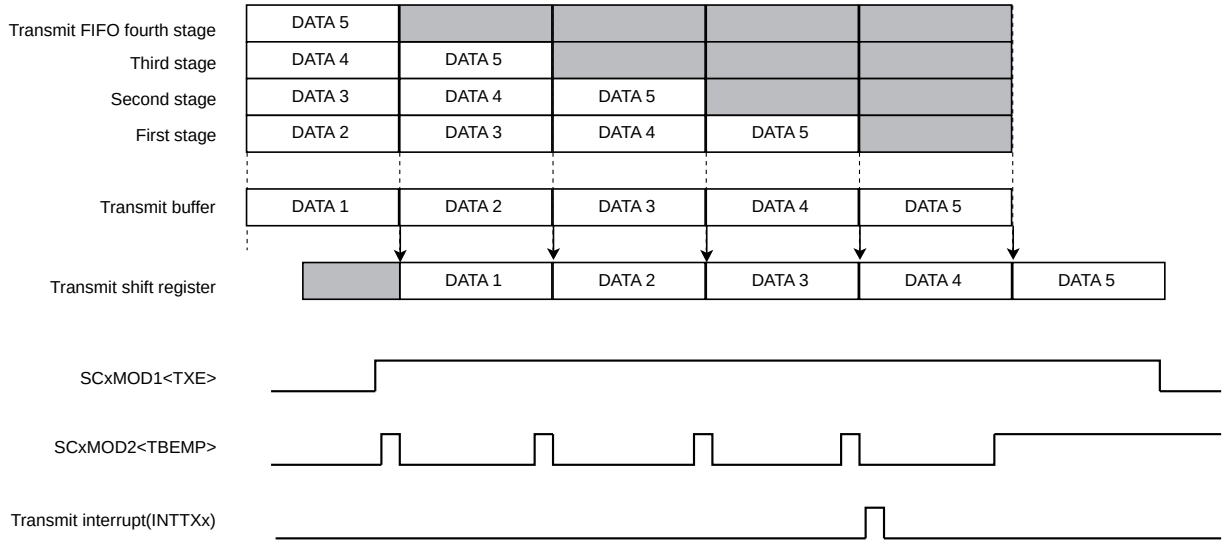
Note: To use Transmit FIFO buffer, Transmit FIFO must be cleared after setting the SIO transfer mode (half duplex/ full duplex) and enabling FIFO (SCxFCNF<CNFG>="1").

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.



13.11.3.3 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.

13.11.3.4 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.

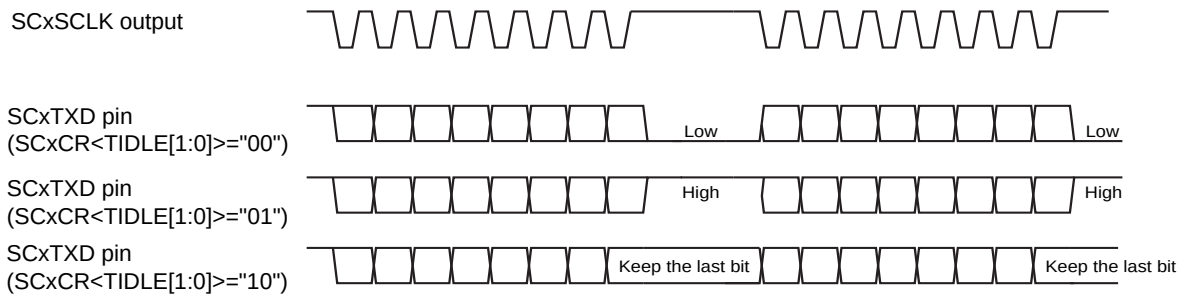


Figure 13-9 Level of SCxTXD pin After the last bit is output

13.11.3.5 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.

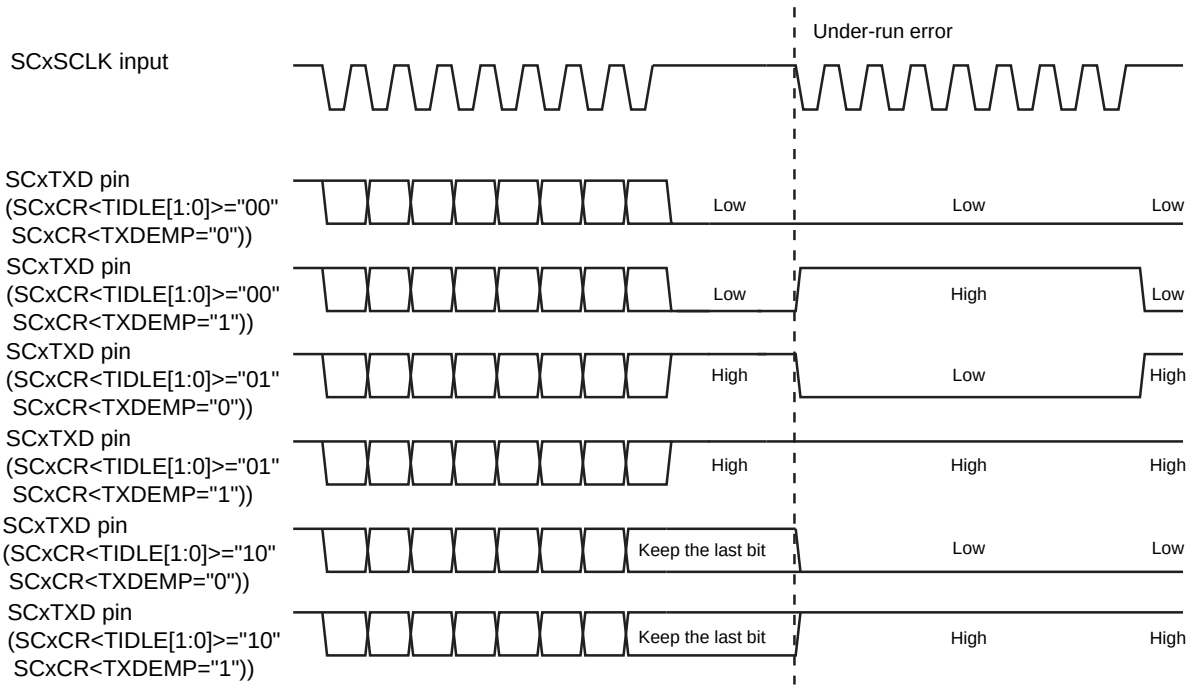


Figure 13-10 Level of SCxTXD pin when Under-run Error is Occurred

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.

13.11.3.6 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

13.12 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.

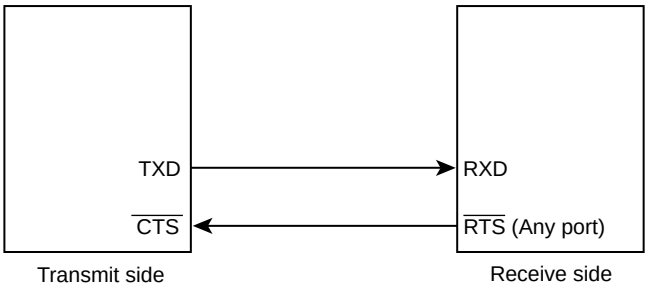


Figure 13-11 Handshake Function

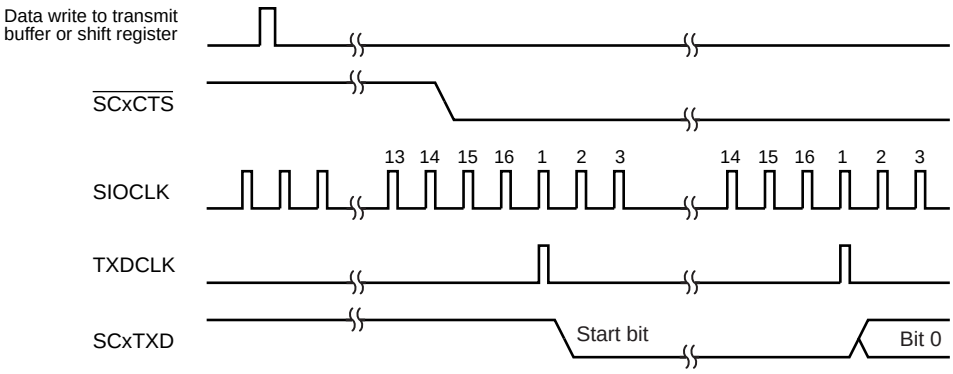


Figure 13-12 $\overline{\text{SCxCTS}}$ Signal timing

13.13 Interrupt/Error Generation Timing

13.13.1 Receive Interrupts

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

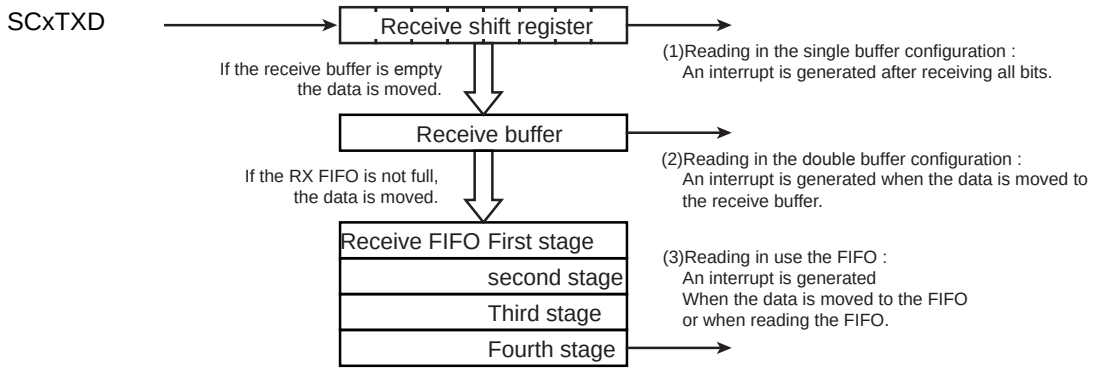


Figure 13-13 Receive Buffer/FIFO Configuration Diagram

13.13.1.1 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 13-7 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 buffter.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.

13.13.1.2 FIFO

When the FIFO is used, a receive interrupt occurs on depending on the timing described in Table 13-8 and the condition specified with SCxRFC<RFIS>.

Table 13-8 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

13.13.2 Transmit interrupts

Figure 13-14 shows the data flow of transmit operation and the route of read.

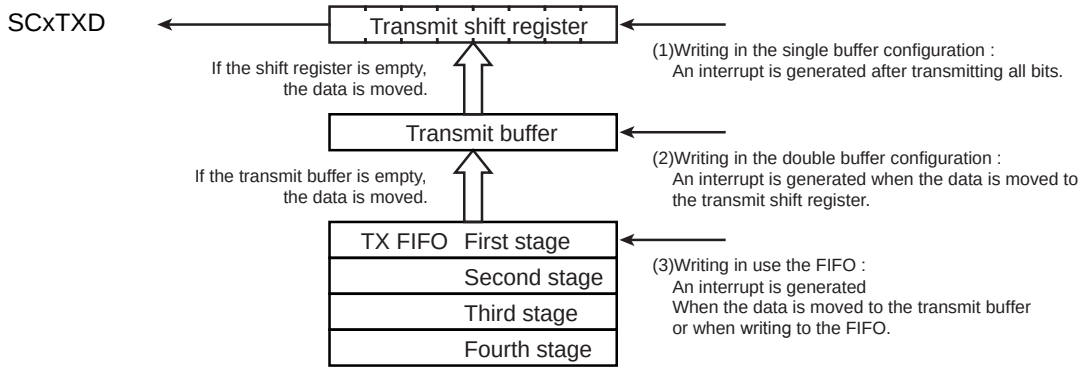


Figure 13-14 Transmit Buffer / FIFO Configuration Diagram

13.13.2.1 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 13-9 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.	

13.13.2.2 FIFO

When the FIFO is used, a transmit interrupt occurs depending on the timing described in Table 13-10 and the condition specified with SCxTFC<TFIS>.

Table 13-10 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

13.13.3 Error Generation

13.13.3.1 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

13.13.3.2 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.

13.14 DMA Request

DMA transfer can be started at the timing of interrupt request.

Please refer to the chapter of "product information" for the channel which can be used for a DMA request with this product.

Note 1: In case using DMA transfer by transmit or receive interrupt request, enabled DMA and set transmit and receive registers after generating software reset by SCxMOD<SWRST>.

Note 2: When the DMA transfer is used, the FIFO cannot be used.

13.15 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.

13.16 Operation in Each Mode

13.16.1 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.

13.16.1.1 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.

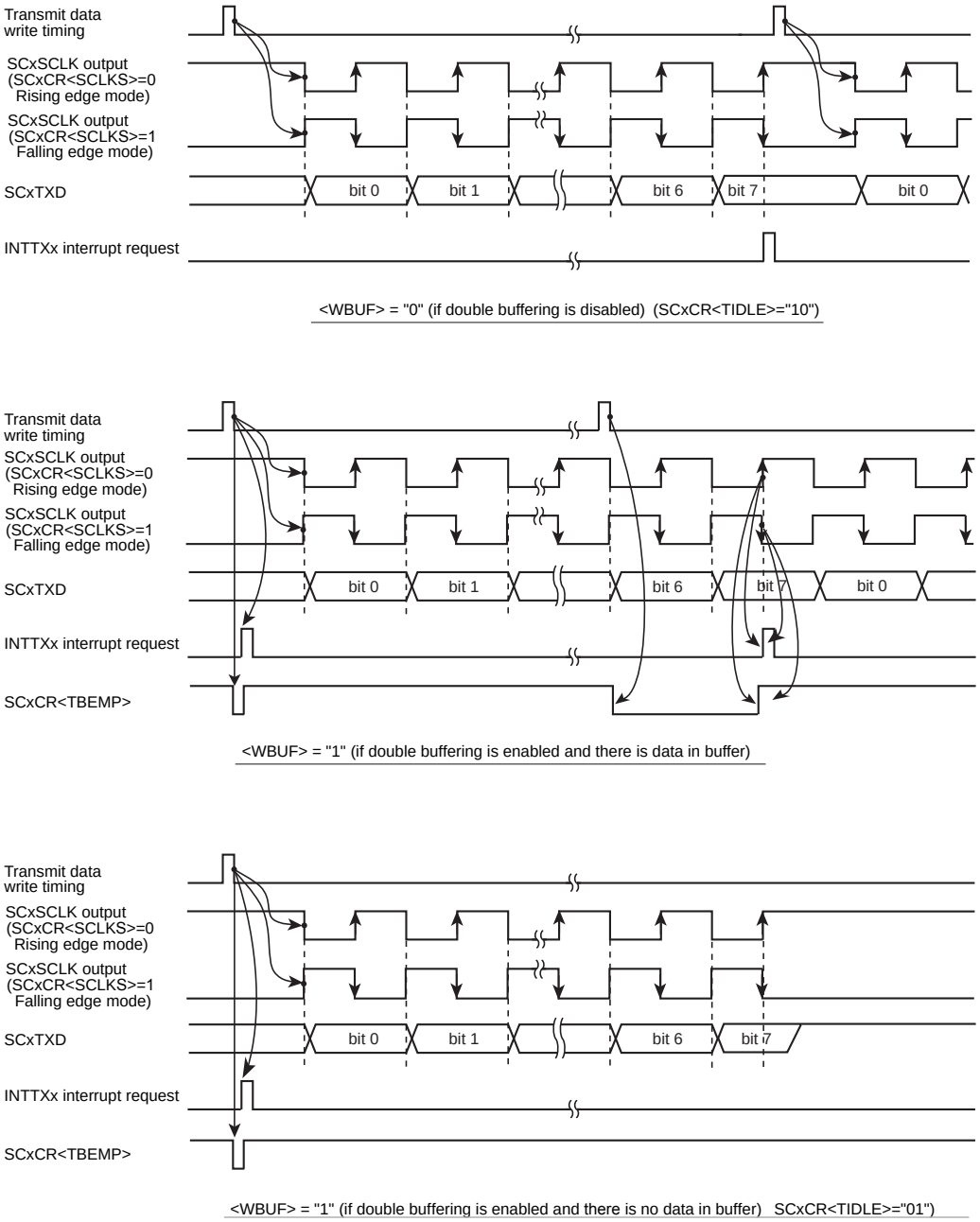


Figure 13-15 Transmit Operation in the I/O Interface Mode (Clock Output Mode)

(2) Clock Input Mode

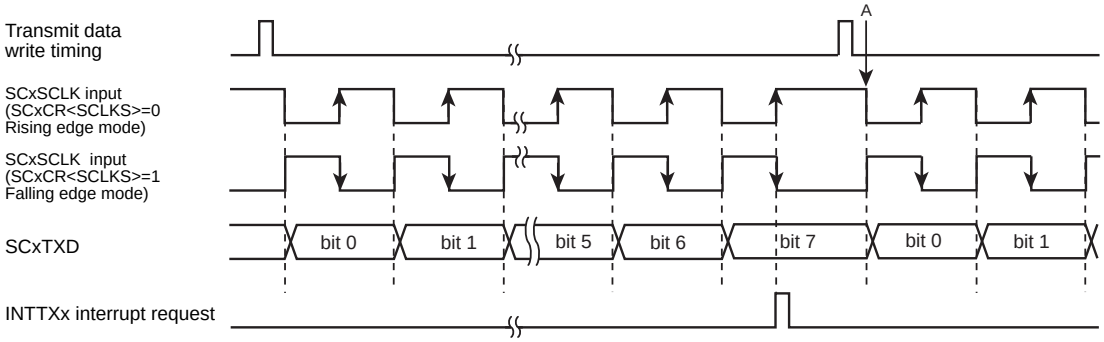
- If double buffering is disabled (SCxMOD2<WBUF> = "0")

  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 13-16.

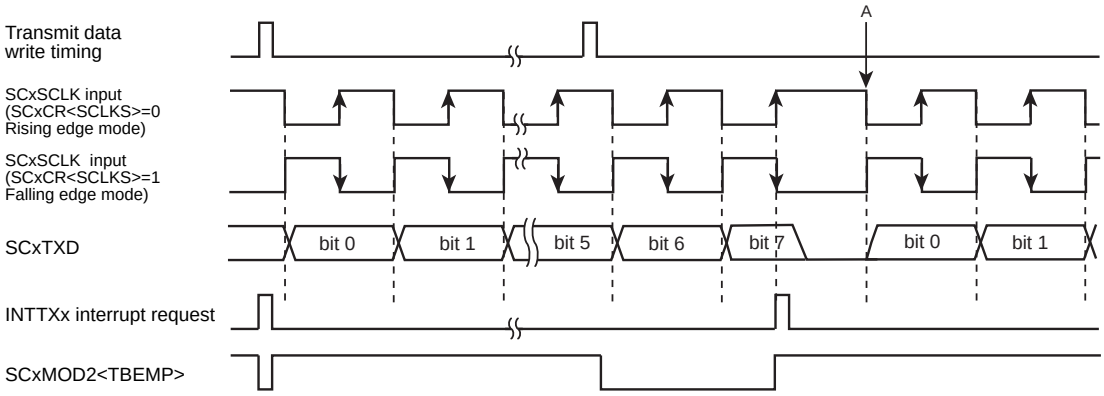
- If double buffer is enabled (SCxMOD2<WBUF> = "1")

  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.

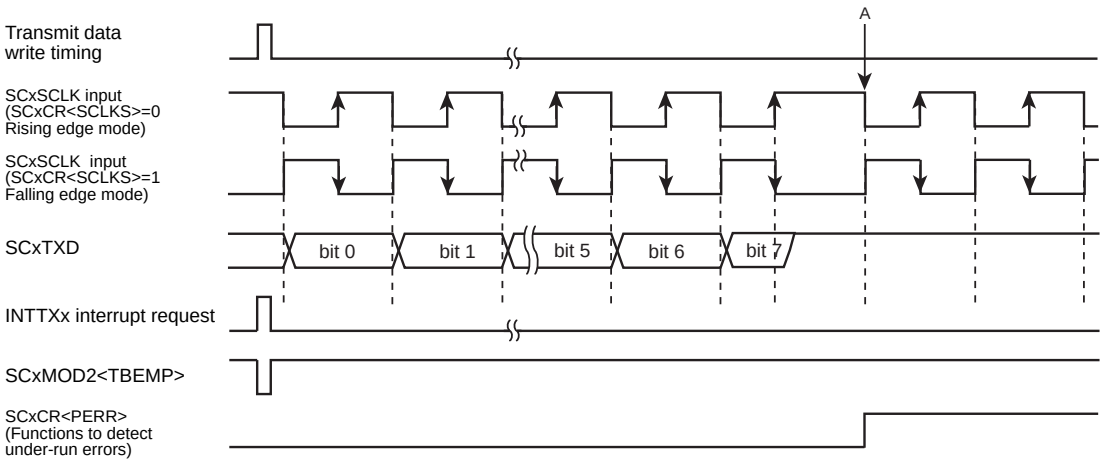
  If the clock input becomes active while no data is in the transmit buffer, although the internal bit counter is started, an under-run error occurs and the level which is specified by SCxCR<TXDEMP> is output to SCxTXD pin.



<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 13-16 Transmit Operation in the I/O Interface Mode (Clock Input Mode)

13.16.1.2 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.

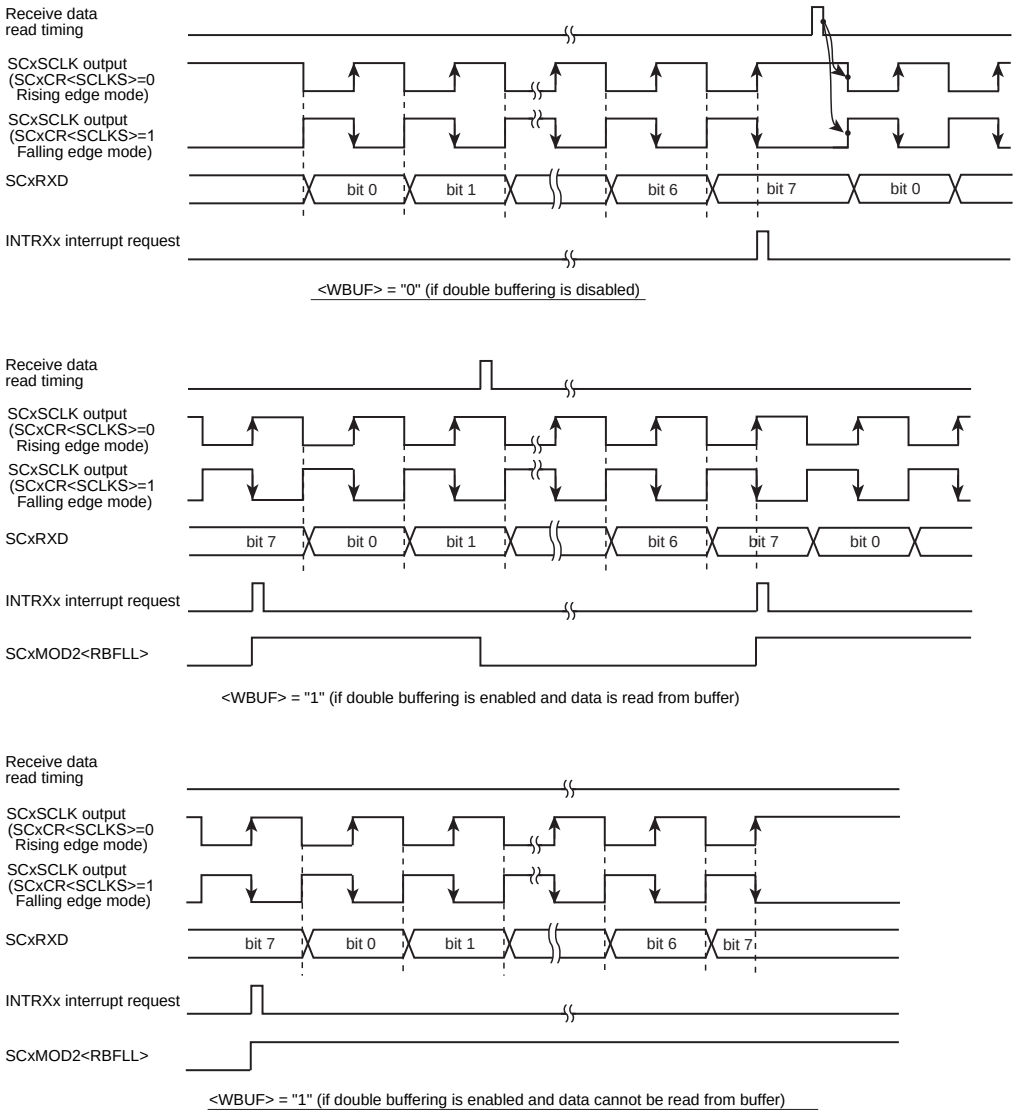


Figure 13-17 Receive Operation in the I/O Interface Mode (Clock Output Mode)

(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.

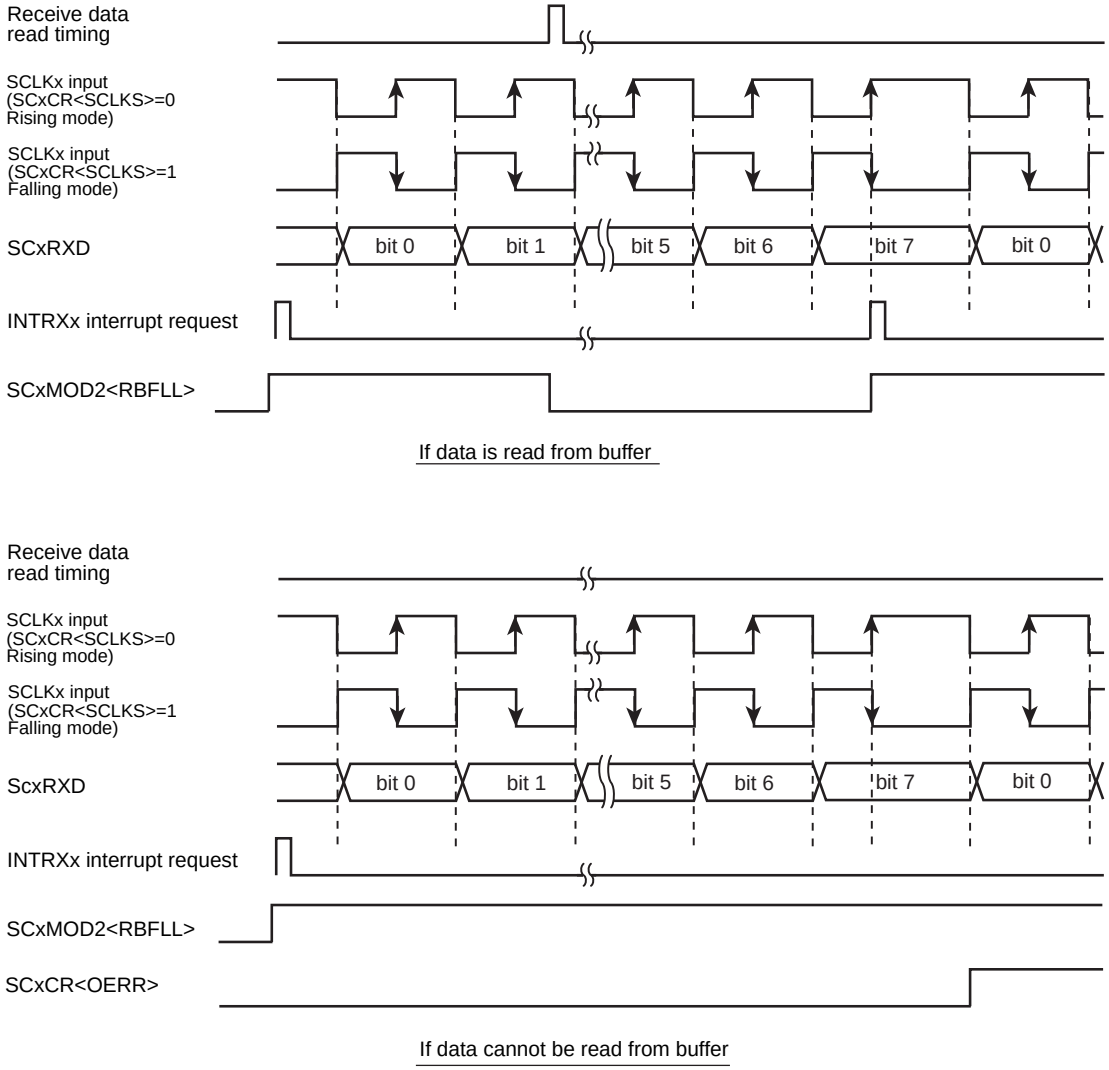


Figure 13-18 Receive Operation in the I/O Interface Mode (Clock Input Mode)

13.16.1.3 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.

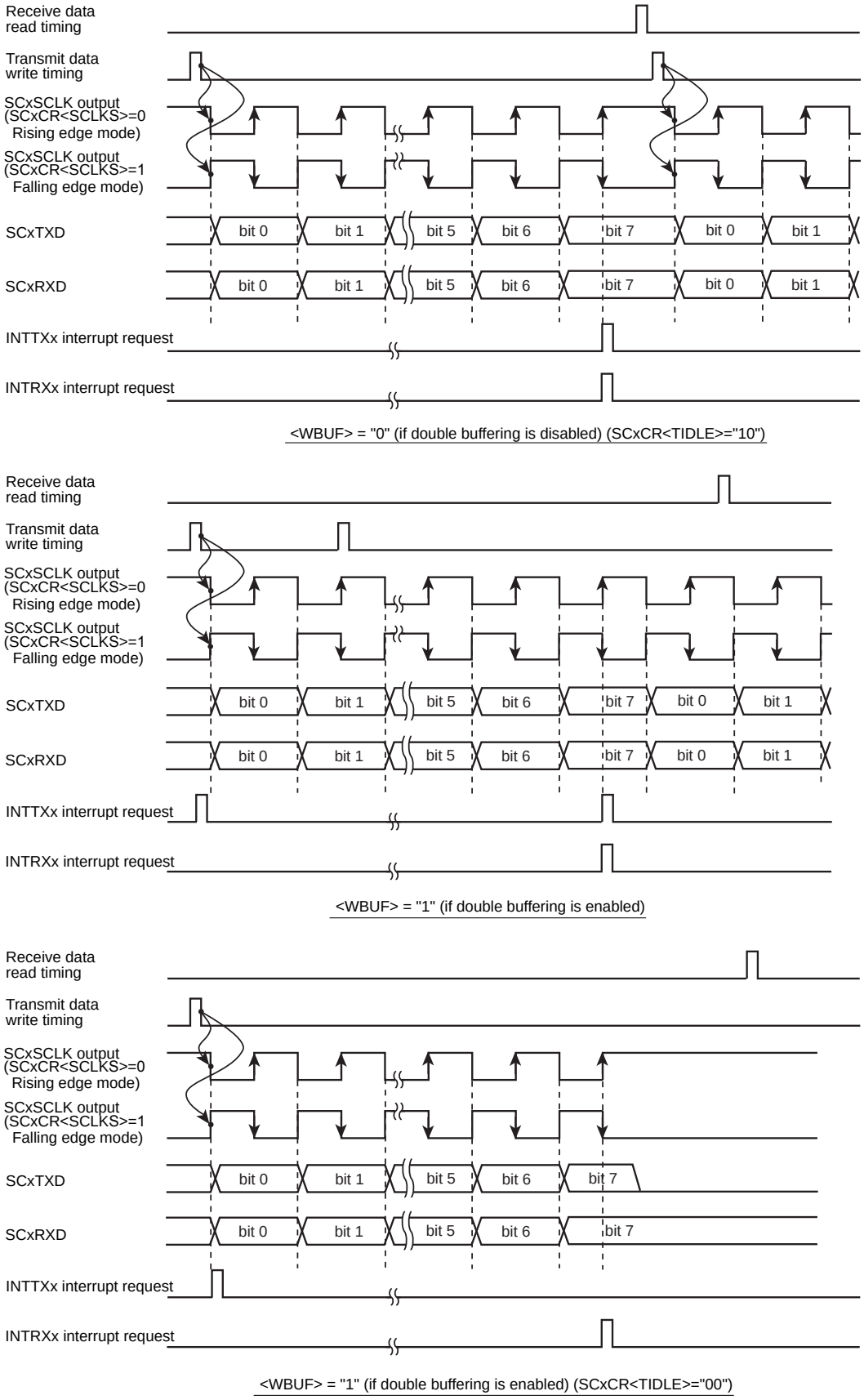


Figure 13-19 Transmit/Receive Operation in the I/O Interface Mode (SCLK Output Mode)

(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 13-20). 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 13-20). 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.

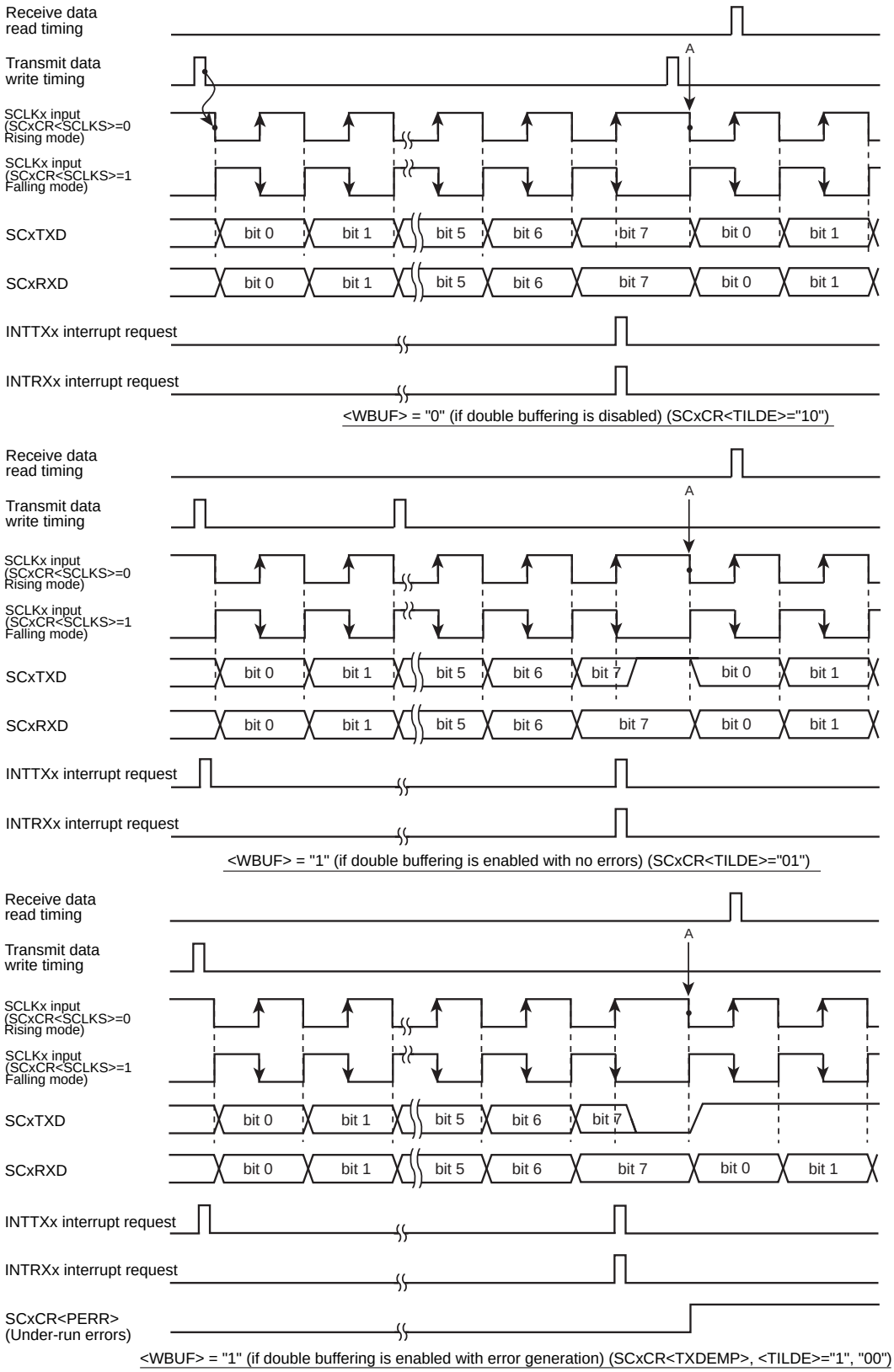


Figure 13-20 Transmit/Receive Operation in the I/O Interface Mode (Clock Input Mode)

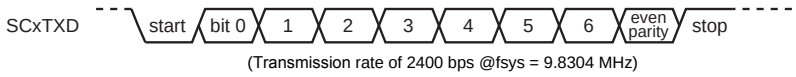
13.16.2 Mode 1 (7-bit UART mode)

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

13.16.3 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

13.16.4 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>.

13.16.4.1 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.

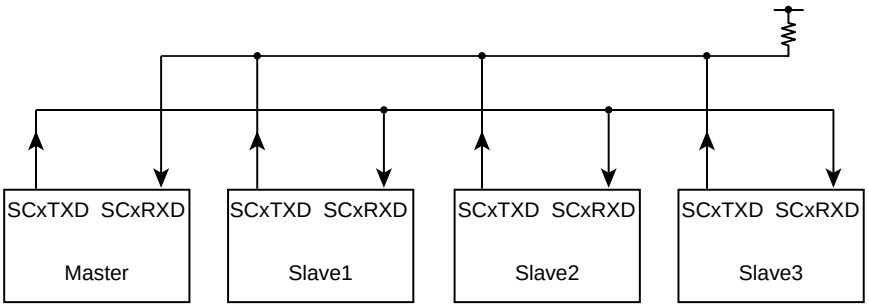
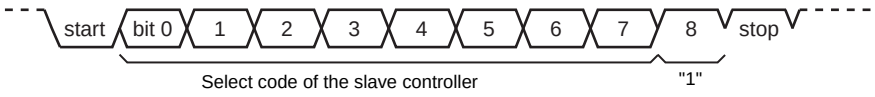


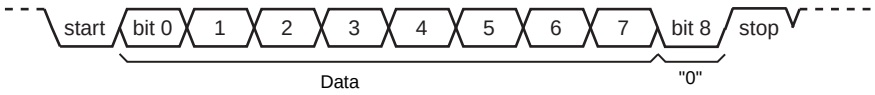
Figure 13-21 Serial Links to Use Wake-up Function

13.16.4.2 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.

14. Serial Bus Interface (I2C/SIO)

The TMPM475FDFG/FZFG/FYFG 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.

14.1 Configuration

The configuration of Serial bus interface is shown in Figure 14-1.

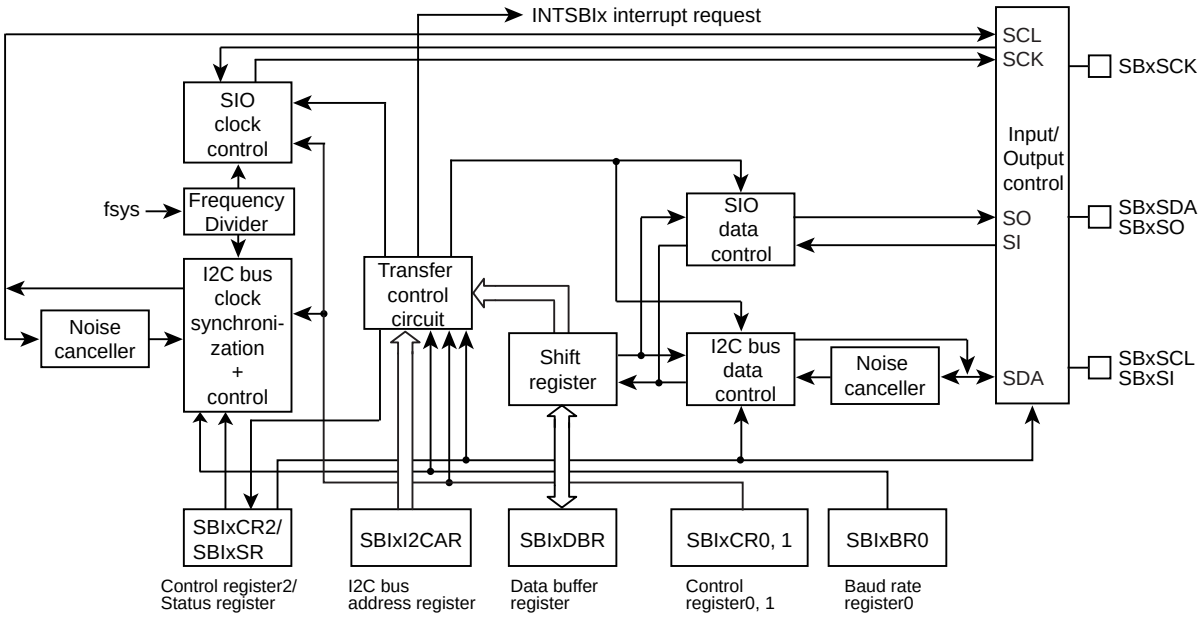


Figure 14-1 Serial Bus Interface Block Diagram

14.2 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 "14.3.1 Control Registers in the I2C Bus Mode" and "14.4.1 Control register of SIO mode".

14.2.1 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

14.3 I2C Bus Mode

14.3.1 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.

14.3.1.1 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.

14.3.1.2 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)

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7-5	BC[2:0]	R/W	Select the number of bits per transfer (Note 1)
4	ACK	R/W	Master mode 0: Acknowledgement clock pulse is not generated. 1: Acknowledgement clock pulse is generated. Slave mode 0: Acknowledgement clock pulse is not counted. 1: Acknowledgement clock pulse is counted.
3	-	R	Read as 1.
2-1	SCK[2:1]	R/W	Select internal SCL output clock frequency (Note 2).
0	SCK[0]	W	
	SWRMON	R	On reading <SWRMON>: Software reset status monitor 0:Software reset operation is in progress. 1:Software reset operation is not in progress.

<BC>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

000	n = 5	System Clock: fsys Clock gear : fc/1 Frequency = $\frac{fsys}{2^n + 72}$ [Hz]
001	n = 6	
010	n = 7	
011	n = 8	
100	n = 9	
101	n = 10	
110	n = 11	
111	reserved	

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

Note 2: For details on the SCL line clock frequency, refer to "14.3.2.2 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.

14.3.1.3 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 "14.3.2.16 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.

14.3.1.4 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"

14.3.1.5 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".

14.3.1.6 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.

14.3.1.7 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.)

14.3.2 Control

14.3.2.1 Operating mode

The setting of SBIxCR2<SBIM[1:0]> controls the operating mode. To operate in I2C mode, set <SBIM [1:0]> to "10".

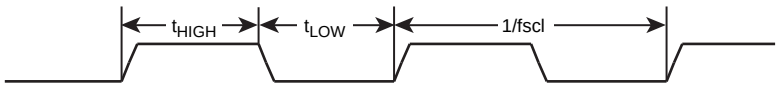
Note 1: Ensure that the SBxSDA pin and SBxSCL pin are at the "High" level before changing to the port mode.

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.

14.3.2.2 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 14-2 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.

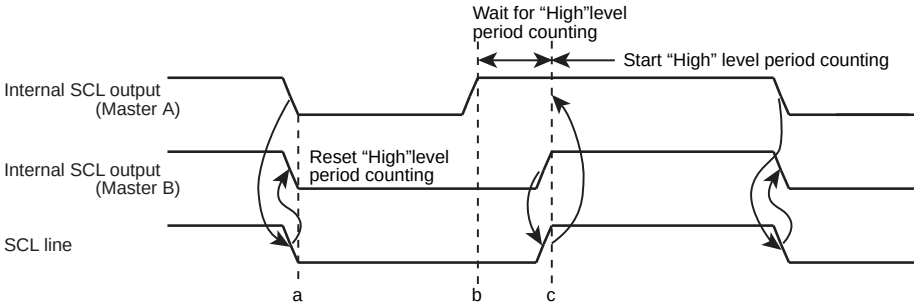


Figure 14-3 Example of Clock Synchronization

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.

14.3.2.3 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.

14.3.2.4 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.

14.3.2.5 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.

14.3.2.6 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.

14.3.2.7 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".

14.3.2.8 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.

14.3.2.9 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>.

14.3.2.10 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.

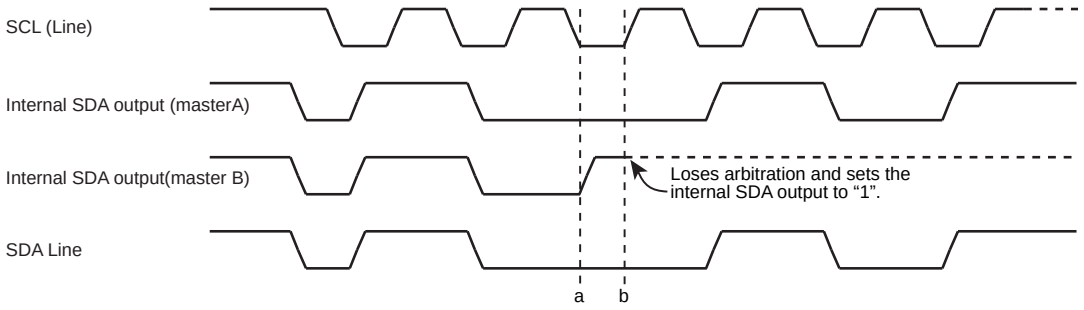


Figure 14-4 Lost Arbitration

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.

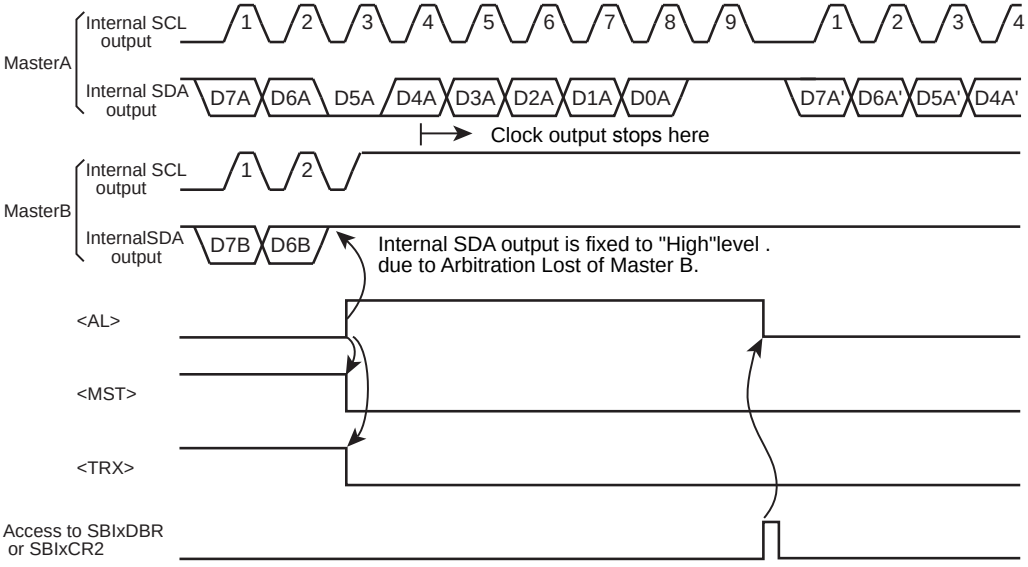


Figure 14-5 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

14.3.2.11 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.

14.3.2.12 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.

14.3.2.13 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.

14.3.2.14 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.

14.3.2.15 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.

14.3.2.16 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.

14.3.3 Data Transfer Procedure

14.3.3.1 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

14.3.3.2 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  1  0  0  0           Generates the start condition.
Example of INTSBIx interrupt routine
    Clears the interrupt request.
    Processing
    End of interrupt
```

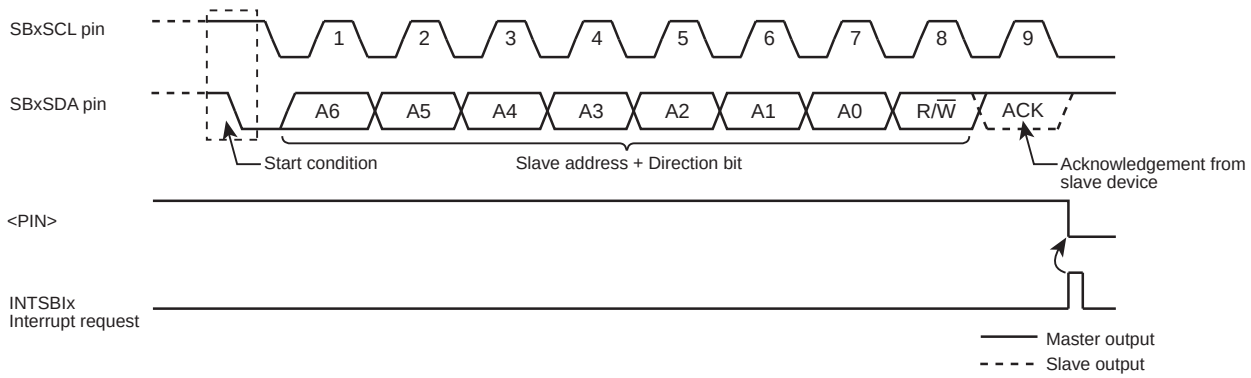


Figure 14-6 Generation of the Start Condition and a Slave Address

14.3.3.3 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**

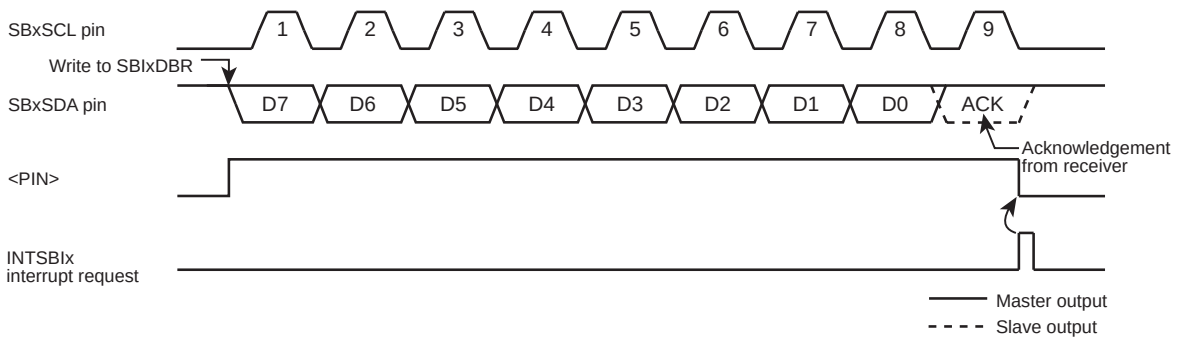


Figure 14-7 <BC[2:0]>= "000",<ACK>= "1" (Transmitter Mode)

(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.

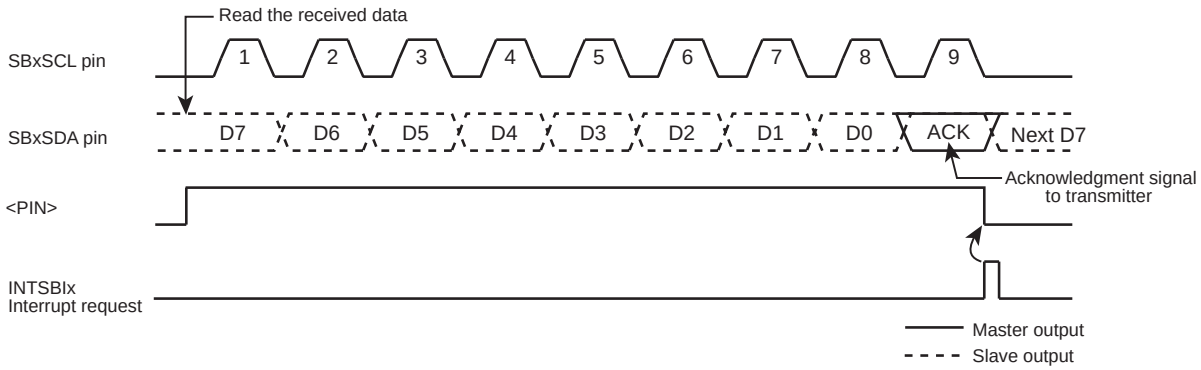


Figure 14-8 <BC[2:0]>= "000",<ACK>= "1" (Receiver Mode)

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.

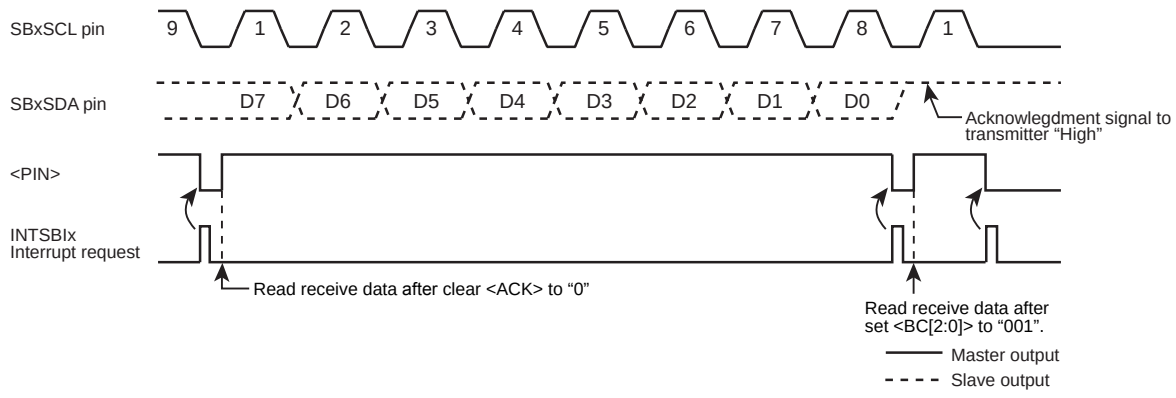


Figure 14-9 Terminating Data Transmission in the Master Receiver Mode

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 14-1 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 14-1 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.

14.3.3.4 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.

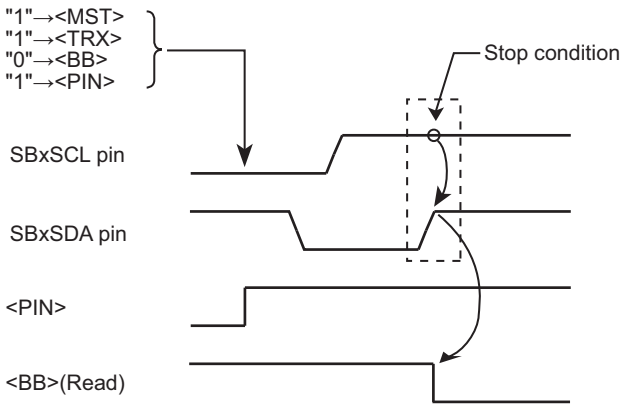


Figure 14-10 Generating the Stop Condition

14.3.3.5 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 "14.3.3.2 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

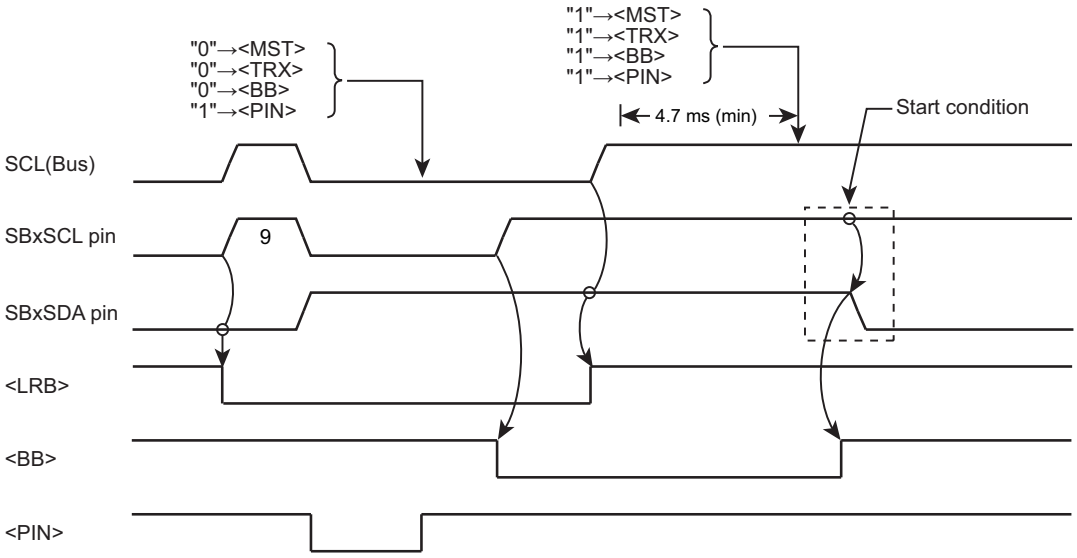
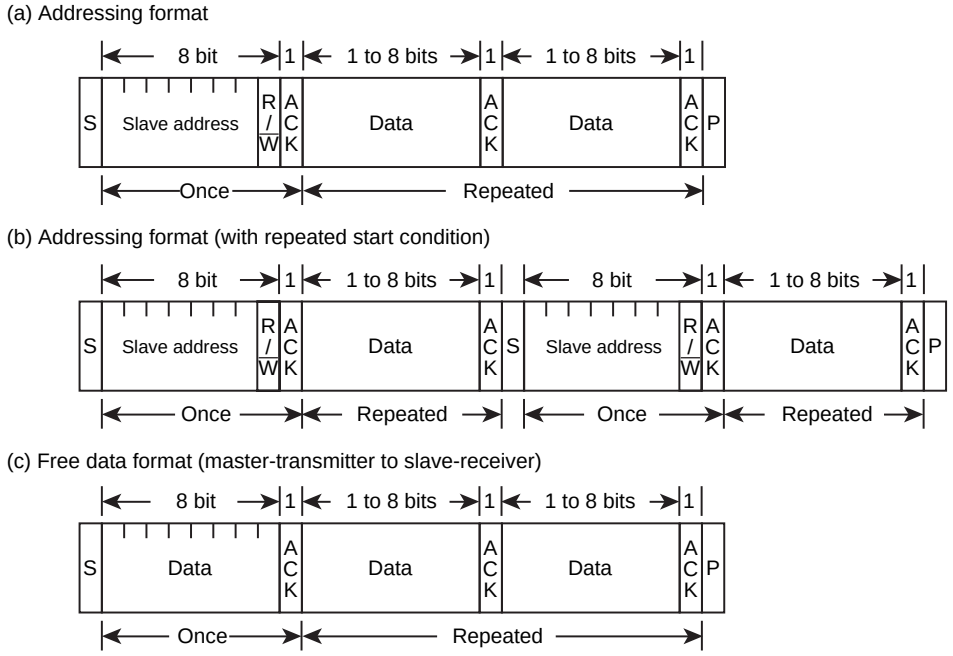


Figure 14-11 Timing Chart of Generating a Restart

14.3.4 Data Format

Figure 14-12 shows the data formats used in the I2C bus mode.



Note) S : Start condition
R/$\overline{W}$: Direction bit
ACK : Acknowledge bit
P : Stop condition

Figure 14-12 I2C Bus Mode Data Formats

14.3.5 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.

14.4 SIO Mode

14.4.1 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.

14.4.1.1 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.

14.4.1.2 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.

14.4.1.3 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.

14.4.1.4 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.

14.4.1.5 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.

14.4.1.6 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".

14.4.2 Control

14.4.2.1 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.

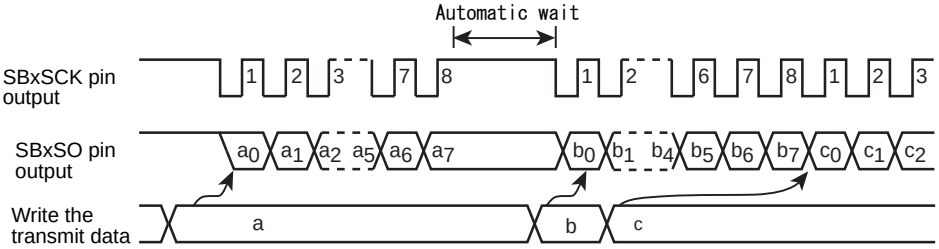
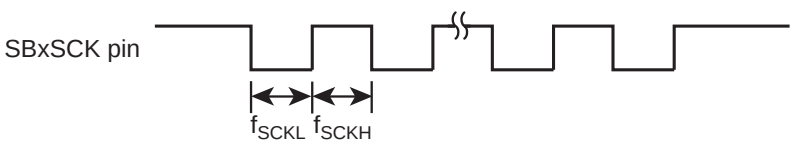


Figure 14-13 Automatic Wait

(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 14-14 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).

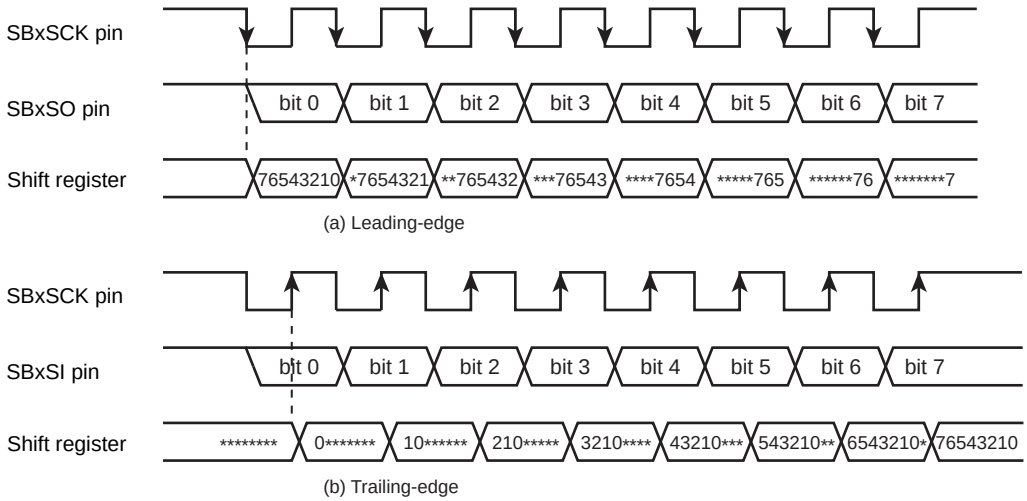


Figure 14-15 Shift Edge

14.4.2.2 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.

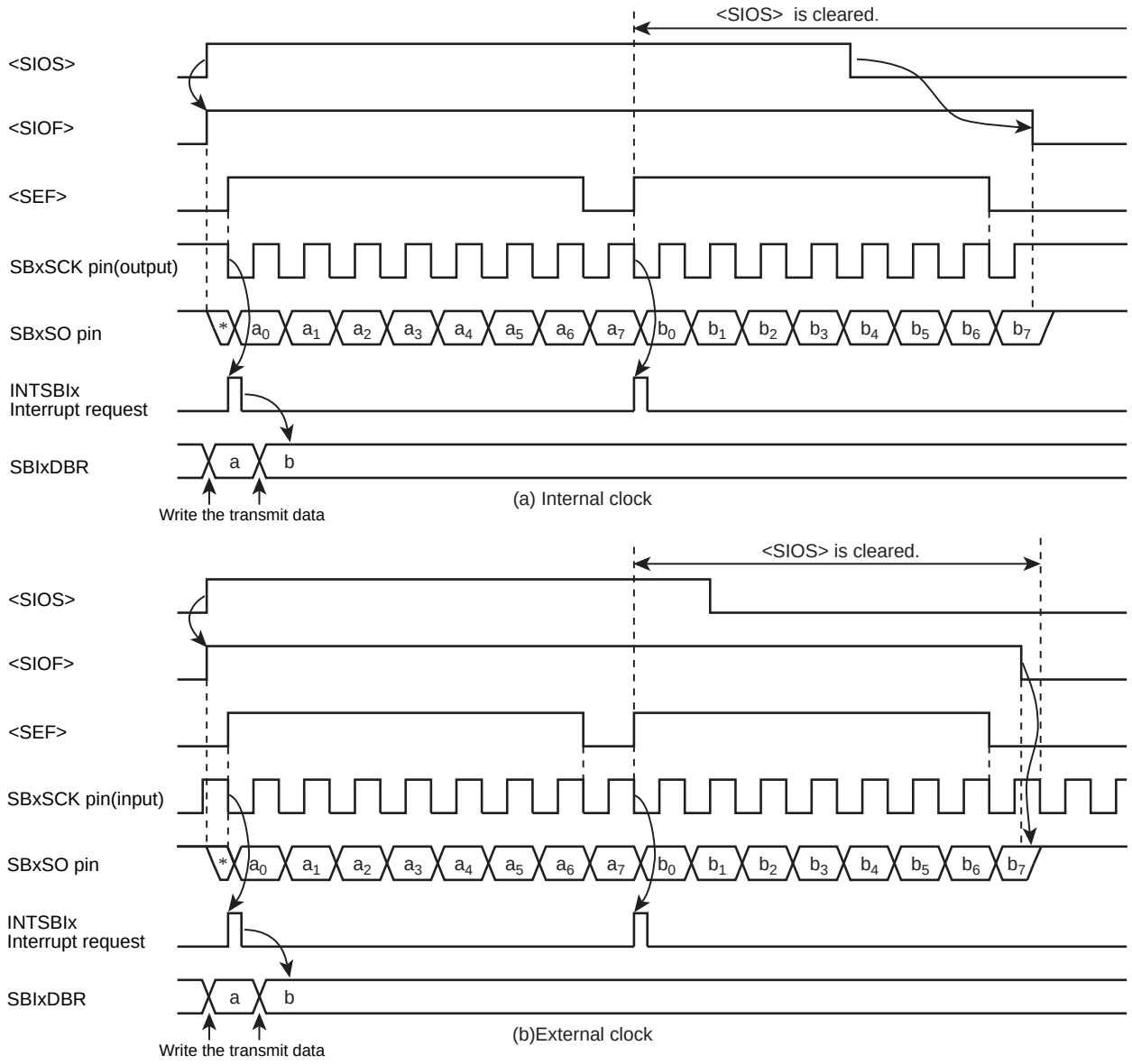


Figure 14-16 Transmit Mode

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.

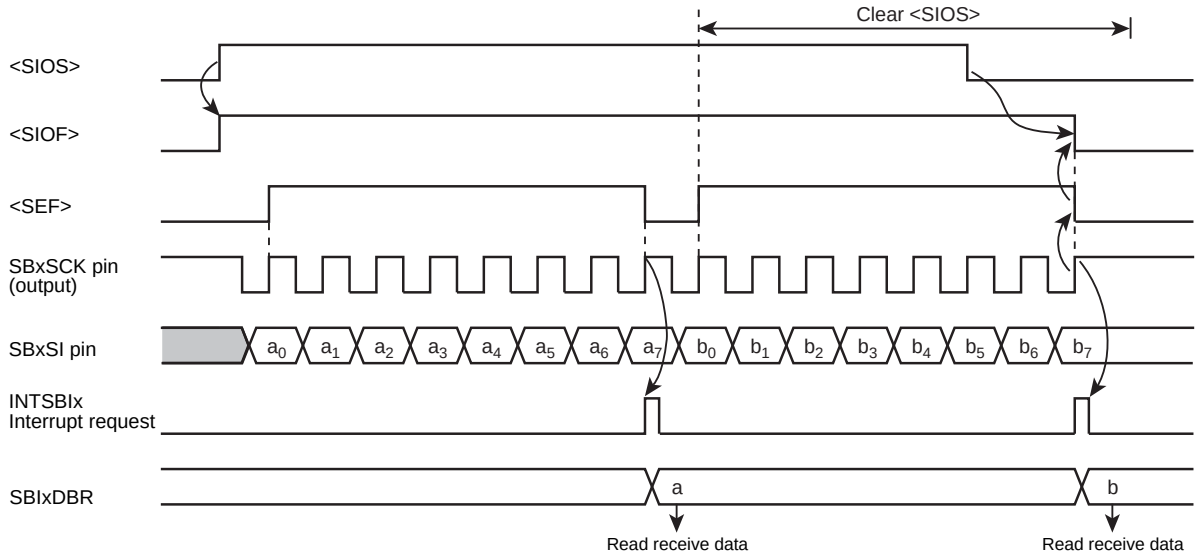


Figure 14-17 Receive Mode (Example: Internal Clock)

(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.

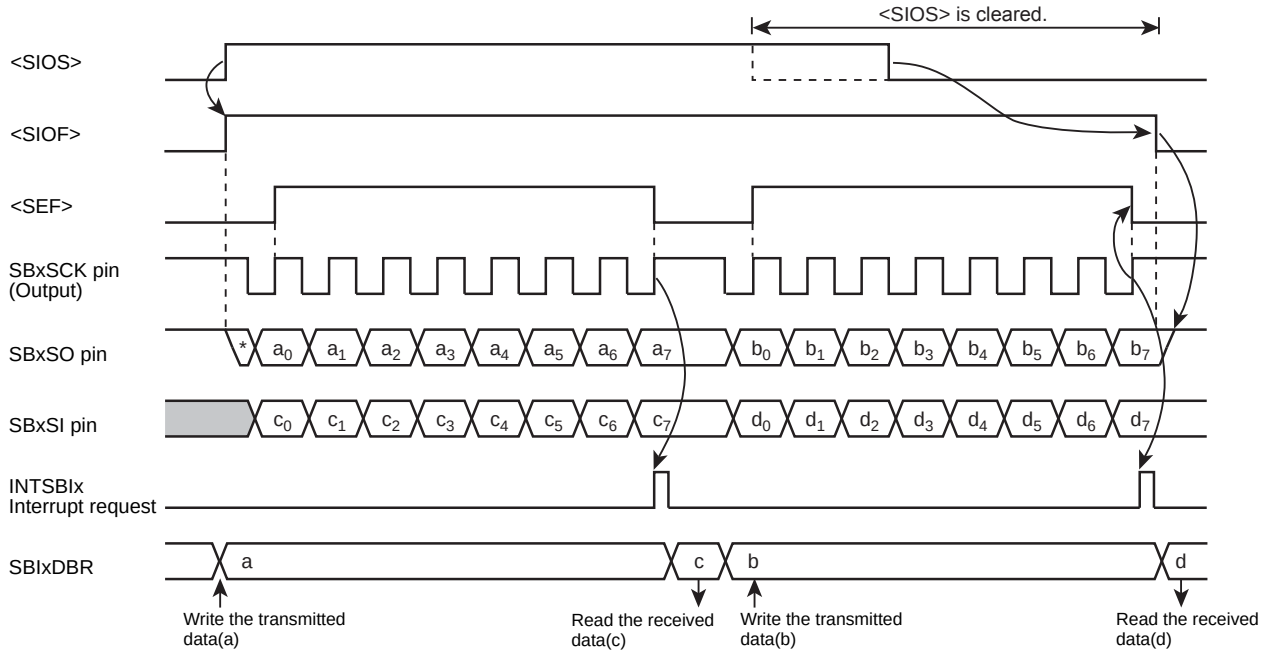


Figure 14-18 Transmit/Receive Mode (Example: Internal Clock)

		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.

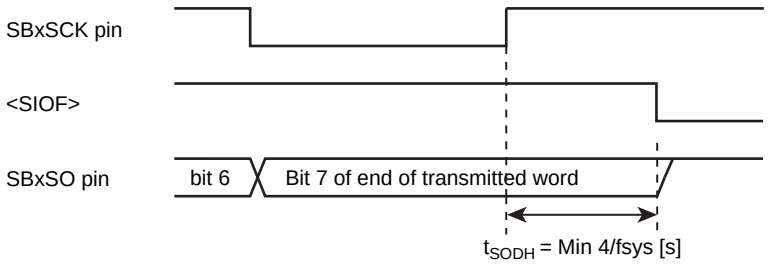


Figure 14-19 Data retention time of the last bit at the end of transmission

15. CAN Controller (CAN)

This product includes one channel of CAN controller.

15.1 Overview

- Compliant with CAN version 2.0 B (active)
- Standard and extended formats supported
- Data frames and remote frames supported for each format
- 32 Mailboxes (31 receive and transmit, 1 receive only)
- CAN bus baud rate up to 1 Mbps (with a system clock of at least 48 MHz)
- Bit timing parameter equivalent to Intel 82527 ™
- Baud rate prescaler built in
- The order in which messages are transmitted can be selected from the following two types of internal arbitrations :
  - The mailbox with the lower number will be sent first
  - The mailbox with the higher priority identifier will be sent first
- Time stamp function for receive and transmit messages
- Operation modes

Normal operation mode	
Configuration mode	
Sleep mode	CAN walk-up with CAN bus active state detection (at CANMCR<WUBA> ="1") or a write access to the master control register MCR
Suspend mode	Inactive state on the CAN bus
Test loop back mode	Self acknowledge
Test error mode	Writable error counters

- Message receive mask function for two systems
  - Programmable global receive mask (common to mailboxes 0 to 31)
  - Programmable local receive mask (for mailbox 31 only)
- Receive mask bit for ID extension bit
- Interrupt signal

INTCANRX	: CAN receive completion interrupt
INTCANTX	: CAN transmit completion interrupt
INTCANGB	: CAN global interrupt Interrupt from eight causes including warning level, error passive and bus-off interrupts)

15.2 Block Diagram

Figure 15-1 shown the block diagram for the CAN controller.

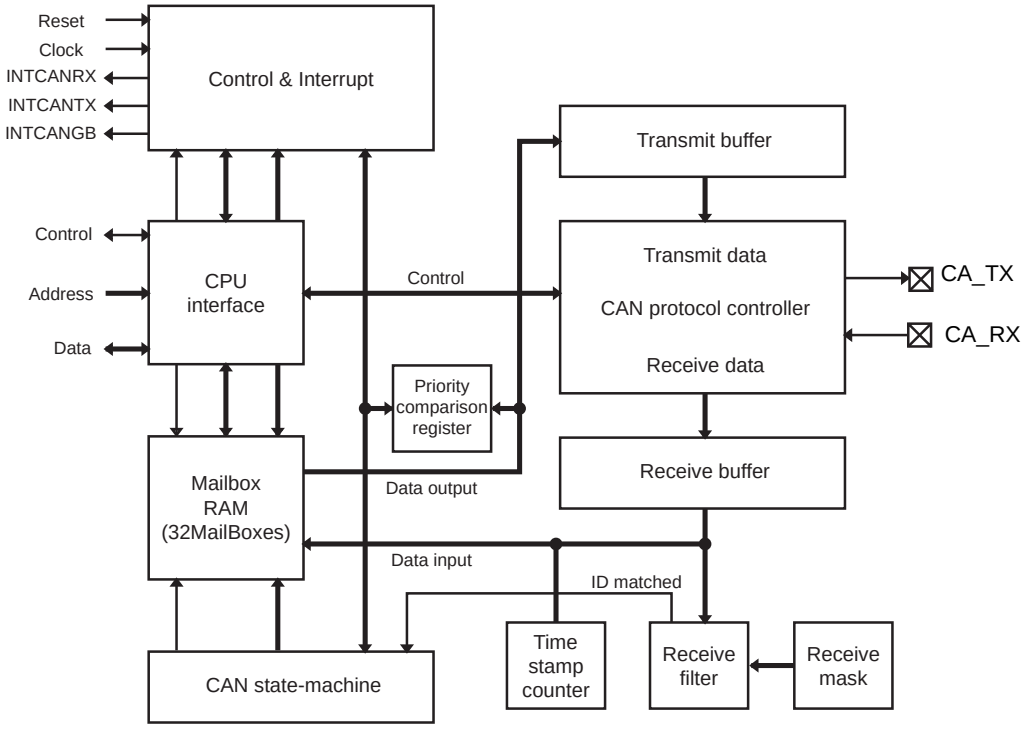


Figure 15-1 Block Diagram of CAN controller

15.3 CAN Interface

The interface to the CAN bus is an input pin CA_RX and an output pin CA_TX. Connect these pins via the CAN bus transceiver (ISO / DIS 11898 compliant).

High speed and low speed transceivers are differentiated. When this IP care must be taken that the electrical characteristics (e.g. , 3.3 V to 5 V) of these pins at chip level satisfy the needs of the transceiver.

15.4 Register

In this MCU, the clock for CAN stops (CGCKSTP<CANSTP> = "0") after reset. When the CAN is used, set the CAN to operate (CGCKSTP<CANSTP> = "1").

For details of the clock operation, refer to the chapter on "Clock/Mode Control".

15.4.1 Register list

Register name (x=0 to 31)		Address(Base+)
Message ID Field Register	CANMBxID	0x0000
Time Stamp Values / Message Control Field Register	CANMBxTSVMCF	0x0008
Data Field Register	CANMBxDL	0x0010
Data Field Register	CANMBxDH	0x0018

Register name		Address(Base+)
Mailbox Configuration Register	CANMC	0x0000
Mailbox Direction Register	CANMD	0x0008
Transmission Request Set Register	CANTRS	0x0010
Transmission Request Reset Register	CANTRR	0x0018
Transmission Acknowledge Register	CANTA	0x0020
Abort Acknowledge Register	CANAA	0x0028
Receive Message Pending Register	CANRMP	0x0030
Receive Message Lost Register	CANRML	0x0038
Local Acceptance Mask Register	CANLAM	0x0040
Global Acceptance Mask Register	CANGAM	0x0048
Master Control Register	CANMCR	0x0050
Global Status Register	CANGSR	0x0058
Bit Configuration Register 1	CANBCR1	0x0060
Bit Configuration Register 2	CANBCR2	0x0068
Global Interrupt Flag Register	CANGIF	0x0070
Global Interrupt Mask Register	CANGIM	0x0078
Mailbox Transmit Interrupt Flag Register	CANMBTIF	0x0080
Mailbox Receive Interrupt Flag Register	CANMBRIF	0x0088
Mailbox Interrupt Mask Register	CANMBIM	0x0090
Change Data Request Register	CANCDR	0x0098
Remote Frame Pending Register	CANRFP	0x00A0
CAN Error Counter Register	CANCEC	0x00A8
Time Stamp Counter Prescaler Register	CANTSP	0x00B0
Time Stamp Counter Register	CANTSC	0x00B8

15.4.2 CANMBxID (Message ID Field Register)

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

Bit	Bit Symbol	Type	Function
31	IDE	R/W	ID Extension bit 0: Standard format (11-bit ID) from <ID28> to <ID18> used 1: Extended format (29-bit ID) from <ID28> to <ID0> used Sets the mailbox by selecting whether to receive or transmit the extended format (<IDE>="1") or the standard format (<IDE>="0").
30	GAME / LAME	R/W	Global (GAME) / Local (LAME) acceptance mask enable bit 0: Receive mask is not used for receive filtering. 1: Receive mask is used for receive filtering. <GAME> is the enable bit for the global acceptance mask GAM shared in mailboxes 0 to 30, and <LAME> is the enable bit for the local acceptance mask LAM used only for mailbox 31. When <GAME>=0 or <LAME>=0, the received message are stored in the mailbox only when the receive message ID is the same as the mailbox ID. For transmit mailboxes, the acceptance mask function is not applied. In such case, always set <GAME> to "0".
29	RFH	R/W	Remote frame handling bit (only for transmit mailboxes) 0:Transmit mailboxes do not respond to remote frames. Software must handle remote frames. 1:Transmit mailboxes respond to remote frames. (The <TRS> bit is set.) <RFH> determines whether a mailbox configured as a transmit mailbox will automatically respond to remote frame reception. When the ID of the received remote frame matches the ID of the transmit mailbox where <RFH>="1" and <GAME>="1", this mailbox ID is overwritten with the remote ID, and the mailbox automatically responds the remote frame using the overwritten ID. Handled as data frames in the case of receive mailboxes.(The <RMP> bit and the <RFP> bit are set.)
28-0	ID[28:0]	R/W	Message ID Standard format (11-bit ID) : From <ID28> to <ID18> are used. Extended format (29-bit ID) :From <ID28> to <ID0> are used. For the priority of message IDs, the message ID having most "0"s consecutively starting from the ID's highest bit (<ID28> bit) has the higher priority.

Register the mailbox IDs at the time of initial setup. To change the message ID field or a mailbox after the mailbox is enabled, clear the <MCx> bit in the CANMC register corresponding to the mailbox to "0", and then disable the mailbox for the CAN controller before writing a new ID.

15.4.3 CANMBxTSVMCF (Time Stamp Values / Message Control Field Register)

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

Bit	Bit Symbol	Type	Function
31-16	TSV[15:0]	R/W	Time stamp counter value The 16-bit time stamp counter values read when message have been successfully received or transmitted are stored. No value is set when message reception or transmission fails. For the details of the entire time stamp counter function, Refer to "15.5.6 Time Stamp Function".
15-5	−	R	Read undefined. Write as "0".
4	RTR	R/W	Remote frame transmit request bit. 0:Data frame 1:Remote frame
3-0	DLC[3:0]	R/W	Data length code Sets the data length (number of bytes) of messages

<DLC[3:0]>	Number of bytes	Corresponding data
0000	0 byte	None
0001	1 byte	D0
0010	2 bytes	D0,D1
0011	3 bytes	D0,D1,D2
0100	4 bytes	D0,D1,D2,D3
0101	5 bytes	D0,D1,D2,D3,D4
0110	6 bytes	D0,D1,D2,D3,D4,D5
0111	7 bytes	D0,D1,D2,D3,D4,D5,D6
1000	8 bytes	D0,D1,D2,D3,D4,D5,D6,D7

When <DLC3:0>="1001" or more is set, data length is processed as 8 bytes.

The time stamp values do not need to be initially set.

The message control field needs no initial programming in the case of receive mailboxes. When a received message is stored in the mailbox, <RTR> and <DLC[3:0]> are also stored in the message control field at the same time. The transmit mailboxes need initial setting.

To change the message control field of a transmit mailbox (which is set to <RFH>="1") after enabling the mailbox, clear the CANMC<MCx> bit to "0" and then disable the mailbox for the CAN controller before writing a new <RTR> and <DLC[3:0]>. The message control field of the transmit mailbox set to <RFH>="0" can be changed irrespective of the CANMC<MCx> bit setting, but the user needs to check that the CANTRS<TRSx> bit is "0" before writing a new <RTR> and <DLC[3:0]>.

15.4.4 CANMBxDL/CANMBxDH (Data fields Register)

For transmission, data is transmitted according to the data byte count set in the <DLC[3:0]> of the mailbox.

For reception, the data length code in the received message is copied to the <DLC[3:0]> of the mailbox, and the data byte count only set in the <DLC[3:0]> is made valid.

Mailboxes are readable and writable, but do not write data fields for receive mailboxes. If data fields are written, a mismatch may occur in received data.

To update the data field of a transmit mailbox set to <RFH>="1", set "1" in CANCDR<CDRx> and suspend transmit requests temporarily before writing new data. To update the data field of a transmit mailbox set to <RFH>="0", check that the CANTRS<TRS> bit is "0" before writing new data.

CANMBxDL

	31	30	29	28	27	26	25	24
bit symbol	D3							
After reset								
	23	22	21	20	19	18	17	16
bit symbol	D2							
After reset								
	15	14	13	12	11	10	9	8
bit symbol	D1							
After reset								
	7	6	5	4	3	2	1	0
bit symbol	D0							
After reset								

Bit	Bit Symbol	Type	Function
31-24	D3[7:0]	R/W	Transmitted and received data is stored.
23-16	D2[7:0]	R/W	Transmitted and received data is stored.
15-8	D1[7:0]	R/W	Transmitted and received data is stored.
7-0	D0[7:0]	R/W	Transmitted and received data is stored.

CANMBxDH

	31	30	29	28	27	26	25	24
bit symbol	D7							
After reset								
	23	22	21	20	19	18	17	16
bit symbol	D6							
After reset								
	15	14	13	12	11	10	9	8
bit symbol	D5							
After reset								
	7	6	5	4	3	2	1	0
bit symbol	D4							
After reset								

Bit	Bit Symbol	Type	Function
31-24	D7[7:0]	R/W	Transmitted and received data is stored.
23-16	D6[7:0]	R/W	Transmitted and received data is stored.
15-8	D5[7:0]	R/W	Transmitted and received data is stored.
7-0	D4[7:0]	R/W	Transmitted and received data is stored.

15.4.5 CANMC (Mailbox Configuration Register)

	31	30	29	28	27	26	25	24
bit symbol	MC31	MC30	MC29	MC28	MC27	MC26	MC25	MC24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MC23	MC22	MC21	MC20	MC19	MC18	MC17	MC16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	MC31 to MC0	R/W	Access configuration to mailbox (Each bit corresponds with mailboxes 31 to 0) 0:The corresponding mailbox MBx is disabled for the CAN controller. 1:The corresponding mailbox MBx is enabled for the CAN controller. Write access from CPU

	ID field	Transmit mailbox with <RFH>="1"	Data field	Control field
<MCx>=0	Enabled	Enabled	Enabled	Enabled
<MCx>=1	Disabled	Disabled	Enabled	Enabled

Note: Following care is required during reprogramming of a CANMC in operation.

Receive: For a receive mailbox it needs to be ensured that the mailbox is not being disabled while reception for this mailbox is ongoing. If a mailbox is disabled or reconfigured during an ongoing reception, the current frame might be received.

Transmit: When the CAN controller is transmitting data (CANTRS<TRSx>="1"), Clear <MCx> to "0" after the transmission is completed (CANTRS<TRSx>="0").

15.4.6 CANMD (Mailbox Direction Register)

	31	30	29	28	27	26	25	24
bit symbol	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24
After reset	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	MD31	R	Mailbox direction : Mailbox 31 Mailbox 31 is the receive-only mailbox. This is always set to "1" and cannot be changed.
30-0	MD30 to MD0	R/W	Mailbox direction : Mailboxes 30 to 0 (Each bit corresponds with mailboxes 30 to 0.) 0:Set as a transmit mailbox. 1:Set as a receive mailbox. Each mailbox can be set as a transmit or receive mailbox.

Set the CANMD register at the initial setup. The directions of mailboxes cannot be changed when operation is ongoing. To change CANMD register settings, set the corresponding CANMC<MCx> bit to "0" before making changes.

15.4.7 CANTRS (Transmission Request Set Register)

	31	30	29	28	27	26	25	24
bit symbol	-	TRS30	TRS29	TRS28	TRS27	TRS26	TRS25	TRS24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TRS23	TRS22	TRS21	TRS20	TRS19	TRS18	TRS17	TRS16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRS15	TRS14	TRS13	TRS12	TRS11	TRS10	TRS9	TRS8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRS7	TRS6	TRS5	TRS4	TRS3	TRS2	TRS1	TRS0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read : Read as "0". Write : Write as "0".
30-0	TRS30 to TRS0	R/W	Transmit request set (Each bit corresponds with mailboxes 30 to 0.) Set <TRSx> requests the message transmission of corresponding mailbox x. When transmission is requested for multiple mailboxes, the message are transmitted in accordance with the priority corresponding to the MCR<MTOS> bit. A write of "1" from the CPU to mailbox x configured as transmit mailbox can set the bit. A write of "0" from the CPU is invalid.

Note: Mailbox 31 is receive-only mailbox.

The transmission request set register can be set by a write of "1" from the CPU to only the CANTRS<TRSx> bits of the mailboxes configured for transmission. The CANTRS<TRSx> bits of the mailboxes configured for reception cannot be set.

The CANTRS<TRSx> bit is cleared to "0" when the message has been successfully transmitted or the transmit request is reset by setting the CANTRR<TRRx> bit to "1."

When transmission fails, the transmission process is repeated until it succeeds or the transmit request is reset by setting the CANTRR<TRRx> bit to "1."

When the CANTRS<TRSx> bit is "1", do not write to mailbox x.

15.4.8 CANTRR (Transmission Request Reset Register)

	31	30	29	28	27	26	25	24
bit symbol	-	TRR30	TRR29	TRR28	TRR27	TRR26	TRR25	TRR24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TRR23	TRR22	TRR21	TRR20	TRR19	TRR18	TRR17	TRR16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRR15	TRR14	TRR13	TRR12	TRR11	TRR10	TRR9	TRR8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read : Read as "0". Write : Write as "0".
30-0	TRR30 to TRR0	R/W	Transmit request reset (Each bit corresponds with mailboxes 30 to 0.) Setting <TRRx> cancels the message transmission of corresponding mailbox x. A write of "1" from the CPU to mailbox x configured as transmit mailbox can set the bit. Write of "0" from the CPU is invalid.

Note: Mailbox 31 is receive-only mailbox.

The transmission request reset register can be set by a write of "1" from the CPU to only the CANTRR<TRRx> bits of the mailboxes configured for transmission. The CANTRR<TRRx> bits of the mailboxes configured for reception cannot be set.

The CANTRR<TRRx> bit is cleared to "0" by the internal logic when the message has been successfully transmitted or the transmission is aborted. A write of "0" from the CPU is invalid.

When the CANTRR<TRRx> bit is "1," do not write to mailbox x.

Setting the CANTRR<TRRx> bit cancels the message transmission of mailbox x set by the CANTRS<TRSx> bit, where the operation executed will be any of the following three sequences:

a. A transmission request of a message has not yet been transmitted.

   A transmission request of a message will be cleared immediately.

   (CANTRS<TRSx> = 0, CANTRR<TRRx> = 0, CANAA<AAx> = 1)

b. A transmission request of a message is currently being transmitted and an arbitration lost error occurs or an error is detected on the CAN bus.

   A transmission request of a message will be cleared and the transmission will be canceled.

   (CANTRS<TRSx> = 0, CANTRR<TRRx> = 0, CANAA<AAx> = 1)

c. A transmission request of a message is currently being transmitted and no arbitration lost error occurs and no error is detected on the CAN bus.

   A transmission request of a message will not be cleared and the transmission will be completed.

   (CANTRS<TRSx> = 0, CANTRR<TRRx> = 0, CANTA<TAx> = 1)

15.4.9 CANTA (Transmission Acknowledge Register)

	31	30	29	28	27	26	25	24
bit symbol	-	TA30	TA29	TA28	TA27	TA26	TA25	TA24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TA23	TA22	TA21	TA20	TA19	TA18	TA17	TA16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TA15	TA14	TA13	TA12	TA11	TA10	TA9	TA8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read : Read as "0". Write : Write as "0".
30-0	TA30 to TA0	R/W	Transmission acknowledge (Each bit corresponds with mailboxes 30 to 0) When the message in mailbox x has been successfully transmitted, the <TAx> bit is set to "1". The <TAx> bit can be cleared by a write of "1" from the CPU to the <TAx> bit or the TRS<TRSx> bit.

Note: Mailbox 31 is receive-only mailbox.

The CANTA<TAx> bit is set to "1" when a message in mailbox x has been successfully transmitted. When the mailbox interrupt is enabled by setting the corresponding <MBIMx> bit in the mailbox interrupt mask register CANMBIM to "1", the <MBTIFx> bit of the mailbox transmit interrupt flag register CANMBTIF is set to "1" and the CAN transmit completion interrupt INTCANTX occurs.

A write of "1" to the <TAx> bit or the CANTRS<TRSx> bit from the CPU can clear the <TAx> bit. A write of "0" to the <TAx> bit or the CANTRS<TRSx> bit from the CPU is invalid.

15.4.10 CANAA (Abort Acknowledge Register)

	31	30	29	28	27	26	25	24
bit symbol	-	AA30	AA29	AA28	AA27	AA26	AA25	AA24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	AA23	AA22	AA21	AA20	AA19	AA18	AA17	AA16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AA15	AA14	AA13	AA12	AA11	AA10	AA9	AA8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AA7	AA6	AA5	AA4	AA3	AA2	AA1	AA0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read : Read as "0". Write : Write as "0".
30-0	AA30 to AA0	R/W	Abort acknowledge (Each bit corresponds with mailboxes 30 to 0.) When the message in mailbox x has not been successfully transmitted, the <AAx> bit is set to "1". The <AAx> bit can be cleared by a write of "1" from CPU to the <AAx> bit or the CANTRS<TRSx> bit.

Note: Mailbox 31 is receive-only mailbox.

The CANAA<AAx> bit is set to "1" when a message in mailbox x has not been successfully transmitted. When CANGIF<TRMABF> bit in the global interrupt flag register is also set to "1", and the transmit abort interrupt is enabled by setting the CANGIM<TRAMABM> bit in the global interrupt mask register to "1", the CAN global interrupt INTCANGB occurs.

A write of "1" to the <AAx> bit or the CANTRS<TRSx> bit from the CPU can clear the <AAx> bit. A write of "0" to the <AAx> bit or the CANTRS<TRSx> bit from the CPU is invalid.

15.4.11 CANRMP (Receive Message Pending Register)

	31	30	29	28	27	26	25	24
bit symbol	RMP31	RMP30	RMP29	RMP28	RMP27	RMP26	RMP25	RMP24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RMP23	RMP22	RMP21	RMP20	RMP19	RMP18	RMP17	RMP16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMP15	RMP14	RMP13	RMP12	RMP11	RMP10	RMP9	RMP8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMP7	RMP6	RMP5	RMP4	RMP3	RMP2	RMP1	RMP0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	RMP31 to RMP0	R/W	Receive message pending (Each bit corresponds with mailboxes 31 to 0.) After a message is received and the content of the received message is written in mailbox x, the <RMPx> bits set to "1". After received data is read, a write of "1" to the <RMPx> bit can clear the <RMPx> bit.

Note:This register cannot use read-modify-write instruction.

The CANRMP<RMPx> bit is set to "1" when a message in mailbox x has been successfully received. When the mailbox interrupt is enabled by setting the corresponding <MBIMx> bit in the mailbox interrupt mask register CANMBIM to "1", the <MBRIFx> bit of the mailbox receive interrupt flag register CANMBRIF is set to "1" and the CAN receive completion interrupt INTCANRX occurs.

To clear the <RMPx> bit, write "1" to the <RMPx> bit from the CPU. A write of "0" to the <RMPx> bit from the CPU is invalid.

15.4.12 CANRML (Receive Message Lost Register)

	31	30	29	28	27	26	25	24
bit symbol	RML31	RML30	RML29	RML28	RML27	RML26	RML25	RML24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RML23	RML22	RML21	RML20	RML19	RML18	RML17	RML16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RML15	RML14	RML13	RML12	RML11	RML10	RML9	RML8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RML7	RML6	RML5	RML4	RML3	RML2	RML1	RML0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	RML31 to RML0	R/W	Receive message lost (Each bit corresponds with mailboxes 31 to 0.) When mailbox x for which the <RMPx> bit is set to "1" receives the next message, the content of the received message is overwritten to the mailbox x, and the <RMLx> bit is set to "1". A write of "1" to the <RMPx> bit can clear the <RMLx> bit.

The CANRML<RMLx> bit is set by the internal logic and can be cleared with a write of "1" to the CANRMP<RMPx> bit from the CPU. The <RMPx> bit is also cleared at the same time. A write of "1" or "0" to the <RMLx> bit from the CPU is invalid.

With the CANRMP<RMPx> bit set to "1", if mailbox x receives the next message, the corresponding <RMLx> bit in the receive message lost register CANRML is set to "1". In this case, mailbox x is overwritten with the new received message.

When the <TRMABF> bit in the global interrupt flag register CANGIF is also set to "1", and the transmit abort interrupt is enabled by setting the <TRMABM> bit in the global interrupt mask register CANGIM to "1", the CAN global interrupt INTCANGB occurs.

When the receive message lost interrupt is enabled by setting the <RMLIM> bit in the global interrupt mask register CANGIM to "1", the CAN global interrupt INTCANGB occurs.

Table 15-1 shows the changes of the CANRMP and CANRML registers before and after a message is received.

Table 15-1 Change of RMP and RML Registers Before / After a Message is Received

ID	Before Reception		After reception		Operation
	<RMPx>	<RMLx>	<RMPx>	<RMLx>	
No match	Don't care	Don't care	Don't care	Don't care	Received message are not stored in any mailboxes.
Match	0	0	1	0	The received message is stored in mailbox x with a matching ID.
	1	0	1	1	The received message is overwritten in mailbox x with a matching ID. This shows that the previous message was lost.
	1	1	1	1	

15.4.13 CANLAM (Local Acceptance Mask Register)

The local acceptance mask register CANLAM will only be used for filtering of the receiving message ID for mailbox 31. This feature allows locally masking to any ID bits of the receiving message for mailbox 31.

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

Bit	Bit Symbol	Type	Function
31	LAMI	R/W	Mask of the <IDE> bit of mailbox 31 0:Not masked 1:Masked In case of <LAMI>="0", the message in the standard or the extended format is received, according to the <IDE> bit of the mailbox 31. In case of <LAMI>="1", the message in the standard and the extended format is received, regardless of the <IDE> bit of the mailbox 31.
30-29	−	R	Read : Read as "0". Write : Write as "0".
28-0	LAM[28:0]	R/W	Mask of receive message ID 0:Not masked The reception message is received when the corresponding bit of reception message ID is the same as mailbox ID. 1:Masked The reception message is received regardless of the value of the corresponding bit of reception message.

In the extended format, < ID[28:0] > and < LAM[28:0] > are used to filtering.

In the standard format, < ID[28:18] > and < LAM[28:18] > are used to filtering.

When the message in a standard format is received, the part of the extended ID (<ID[17:0]>) will become an undefined value. Therefore, the standard and the extended format cannot be recommended to be received in alternately the same mailbox.

Please set CANLAM when initialization (At the configuration mode) and do not change the setting while operating. When the setting is changed while receiving the message, the CANLAM value on the way of the setting change is used to filtering of reception message ID.

15.4.14 CANGAM (Global Acceptance Mask Register)

The global acceptance mask register CANGAM will be used for filtering of the receiving message ID for mailbox 0 to 30. This feature allows to globally masking any ID bits of the receiving message for mailbox 0 to 30.

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

Bit	Bit Symbol	Type	Function
31	GAMI	R/W	Mask of the <IDE> bit of mailboxes 0 to 30 0:Not masked 1:masked In case if <GAMI> =" 0", the message of the standard or the extended format is received, according to the <IDE> bit of the mailboxes 0 to 30. In case of <GAMI> = "1", the message of the standard and the extended format is received, regardless of the <IDE> bit of the mailboxes 0 to 30.
30-29	−	R	Read : Read as "0". Write : Write as "0".
28-0	GAM[28:0]	R/W	Mask of receive message ID 0:Not masked The reception message is received when the corresponding bit of reception message ID is the same as mailbox ID. 1:Masked The reception message is received regardless of the value of the corresponding bit of reception message.

In the extended format, < ID[28:0] > and < GAM[28:0] > are used for filtering.

In the standard format, < ID[28:18] > and < GAM[28:18] > are used for filtering.

When the message in the standard format is received, the part of the extended ID (<ID[17:0]>) will become an undefined value. Therefore, the standard and the extended format cannot be recommended to be received in alternately the same mailbox.

Please set CANGAM during the initialization (At the configuration mode) and do not change the setting during the operation. When the setting is changed while receiving the message, the CANGAM value on the way of the setting change is used for filtering of reception message ID.

15.4.15 CANMCR (Master 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	-	-	-	-	SUR	-	TSTLB	TSTERR
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CCR	SMR	-	WUBA	MTOS	-	TSCC	SRES
After reset	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read : Read as "0". Write : Write as "0".
11	SUR	R/W	Suspend mode request 0: Cancels suspend mode (normal operation) 1: Request suspend mode
10	−	R	Read : Read as "0". Write : Write as "0".
9	TSTLB	R/W	Test loop back 0:Cancels test loop back mode (normal operation) 1:Request test loop back mode (This mode supports stand-alone operation.)
8	TSTERR	R/W	Test error 0:Cancels test error mode (normal operation) 1:Request test error mode (In this mode, it is possible to write the CAN error counter register (CANCEC)).
7	CCR	R/W	Change configuration request 0:Cancels configuration mode (normal operation) 1:Request configuration mode (In this mode, it possible to write the bit configuration registers, CANBCR1 and CANBCR2.)
6	SMR	R/W	Sleep mode request 0:Cancels sleep mode (normal operation) 1:Request sleep mode (In this mode, the clock of the CAN controller stops and the error counters and transmit requests are reset.)
5	−	R	Read : Read as "0". Write : Write as "0".
4	WUBA	R/W	Walk-up on bus activity 0: Wakes up only by a write access to the CAMCR register. 1:Wakes up by detecting a bus active state or a write access to the CAMCR.
3	MTOS	R/W	Mailbox transmission order select 0:Messages are transmitted in ascending order of mailbox number. 1:Messages in mailboxes are transmitted in descending order of message ID priority.
2	−	R	Read : Read as "0". Write : Write as "0".
1	TSCC	R/W	Time stamp counter clear 0: Disable 1:Clears the time stamp counter to "0". (Note1) This bit is for write only and is read as always "0".

Bit	Bit Symbol	Type	Function
0	SRES (Note2)	R/W	Software reset 0:Disable 1:Resets the CAN controller by software. This bit is for write only and is read as always "0".

Note 1: The time stamp counter is also cleared by a write to the CANTSP register and a write of "0" to the CANTSC register.

Note 2: After software reset, all registers in CAN must be access after the following time.

(1) When communication by CAN bus is not performed, please wait more than 16 CPU clocks.

(2) When communication by CAN bus is performed, please wait more than 88 CPU clocks.

Note 3: To cancel sleep mode of the CAN operation, check that the CANGSR<SMA> bit is "1" before setting CANMCR<SMR> to "0".

15.4.16 CANGSR (Global 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	-	-	-	-	-	-	-	MIS
After reset	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	MIS				RM	TM	-	SUA
After reset	1	1	1	1	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CCE	SMA	-	-	TSO	BO	EP	EW
After reset	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-17	−	R	Read : Read as "0". Write : Write as "0".
16-12	MIS[4:0]	R	Message in slot Indicates the mailbox number of a message located in the transmit buffer. 00000 :Message for mailbox 0 00001 :Message for mailbox 1 00010 :Message for mailbox 2 00011 :Message for mailbox 3 00100 :Message for mailbox 4 00101 :Message for mailbox 5 00110 :Message for mailbox 6 00111 :Message for mailbox 7 01000 :Message for mailbox 8 01001 :Message for mailbox 9 01010 :Message for mailbox 10 01011 :Message for mailbox 11 01100 :Message for mailbox 12 01101 :Message for mailbox 13 01110 :Message for mailbox 14 01111 :Message for mailbox 15 10000 :Message for mailbox 16 10001 :Message for mailbox 17 10010 :Message for mailbox 18 10011 :Message for mailbox 19 10100 :Message for mailbox 20 10101 :Message for mailbox 21 10110 :Message for mailbox 22 10111 :Message for mailbox 23 11000 :Message for mailbox 24 11001 :Message for mailbox 25 11010 :Message for mailbox 26 11011 :Message for mailbox 27 11100 :Message for mailbox 28 11101 :Message for mailbox 29 11110 :Message for mailbox 30 11111 : There is no message in the transmit buffer.
11	RM	R	Receive mode 0:The CAN controller is not receiving a message. 1:The CAN controller is receiving a message.
10	TM	R	Transmit mode 0:The CAN controller is not transmitting a message. 1:The CAN controller is transmitting a message.
9	−	R	Read : Read as "0". Write : Write as "0".
8	SUA	R	Suspend mode acknowledge 0:The CAN controller is not in suspend mode. 1:The CAN controller is in suspend mode.
7	CCE	R	Change configuration enable 0:The CAN controller is not in configuration mode. 1:The CAN controller is in configuration mode. In this mode, it is possible to write the bit configuration registers, CANBCR1 and CANBCR2.
6	SMA	R	Sleep mode acknowledge 0:The CAN controller is not in sleep mode. 1:The CAN controller is in sleep mode. In this mode, the clock of the CAN controller stops and the error counters and transmit request are reset.
5-4	−	R	Read : Read as "0". Write : Write as "0".
3	TSO	R	Time stamp overflow 0:The time stamp counter is not overflow. 1:The time stamp counter has overflow at least once after this bit was last cleared to "0". To clear this bit, clear <TSOIF> bit in the CANGIF register to "0".

Bit	Bit Symbol	Type	Function
2	BO	R	Bus off status 0:In bus on state (normal operation) 1:In bus off state When CAN bus errors occur abnormally often and the transmit error counter <TEC> reaches its limit of 256, the CAN controller enters bus off state. No messages can be transmitted and received, The error counter is undefined. After the bus off recovery sequence, the CAN controller automatically enters bus on state.
1	EP	R	Error passive status 0:The CAN controller is not in error passive mode. 1:The CAN controller is in error passive mode.
0	EW	R	Warning status 0:Both <TEC> and <REC> values are 96 or less. 1:At least one of the <TEC> and <REC> values is greater than 96 and has reached the warning level.

15.4.17 CANBCR1 (Bit Configuration 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	-	-	-	-	-	-	BRP	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BRP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-10	−	R	Read : Read as "0". Write : Write as "0".
9-0	BRP[9:0]	R/W	Baud rate prescaler Setting value : 0 to 1023

15.4.18 CANBCR2 (Bit Configuration 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	-	-	-	-	-	-	SJW	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SAM	TSEG2			TSEG1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-10	−	R	Read : Read as "0". Write : Write as "0".
9-8	SJW[1:0]	R/W	Resynchronization jump width. 00 : 1 × TQ 01 : 2 × TQ 10 : 3 × TQ 11 : 4 × TQ
7	SAM	R/W	Setting sampling count 0:Single sampling 1 Triple sampling
6-4	TSEG2[2:0]	R/W	Setting of bit time after sample point 000 : Reserved 100 : 5 × TQ 001 : 2 × TQ 101 : 6 × TQ 010 : 3 × TQ 110 : 7 × TQ 011 : 4 × TQ 111 : 8 × TQ
3-0	TSEG1[3:0]	R/W	Setting of bit time before sample point (except SYNCSEG). 0000 : Reserved 1000 : 9 × TQ 0001 : 2 × TQ 1001 : 10 × TQ 0010 : 3 × TQ 1010 : 11 × TQ 0011 : 4 × TQ 1011 : 12 × TQ 0100 : 5 × TQ 1100 : 13 × TQ 0101 : 6 × TQ 1101 : 14 × TQ 0110 : 7 × TQ 1110 : 15 × TQ 0111 : 8 × TQ 1111 : 16 × TQ

15.4.19 CANGIF (Global Interrupt 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	RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read : Read as "0". Write : Write as "0".
7	RFPF	R/W	Remote frame pending flag 0:No remote frame has been received. 1:Remote frames have been received. (in the receive mailbox) This bit will not be set when matching with the transmit mailbox for which the <RFH> bit is "1".
6	WUIF	R/W	Walk-up interrupt flag 0:In sleep mode or normal operation mode 1:Sleep mode has been canceled.
5	RMLIF	R/W	Receive message lost interrupt flag 0:No receive message lost error has occurred. 1:A receive message lost error has occurred in at least one mailbox configured as a receive mailbox.
4	TRMABF	R/W	Transmission abort flag 0:No transport abort has occurred. 1:Transport abort has occurred. (At least one bit in the CANAA register is set.)
3	TSOIF	R/W	Time stamp counter overflow interrupt flag 0:No overflow has occurred in the time stamp counter after this bit was last cleared. 1:There was at least one overflow of the time stamp counter after this bit was last cleared.
2	BOIF	R/W	Bus off interrupt flag 0: The CAN controller is in bus on mode. 1: The CAN controller is in bus off mode.
1	EPIF	R/W	Error passive interrupt flag 0: The CAN controller is in error active mode. 1: The CAN controller is in error passive mode.
0	WLIF	R/W	Warning level interrupt flag 0:None of the error counters have reached the warning level. 1: At least one of the error counters has reached the warning level.

Each interrupt flag of the global interrupt flag register (CANGIF) will be set to "1" if the corresponding global interrupt condition has met. When the global interrupt flag is set to "1", if the corresponding bit in the global interrupt mask register (CANGIM) is "1" (interrupt enabled), the CAN global interrupt (INTCANGB) will be "High".

The CANGIF register can be cleared by writing "1" to the corresponding bit in the CANGIF register. A write of "0" is invalid.

15.4.20 CANGIM (Global 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	RFPFM	WUIM	RMLIM	TRMABF	TSOIM	BOIM	EPIM	WLIM
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read : Read as "0". Write : Write as "0".
7	RFPM	R/W	Remote frame pending interrupt mask 0:Interrupt disable 1:Interrupt enable
6	WUIM	R/W	Walk-up interrupt mask 0:Interrupt disable 1:Interrupt enable
5	RMLIM	R/W	Receive message lost interrupt mask 0:Interrupt disable 1:Interrupt enable
4	TRMABF	R/W	Transmit abort interrupt mask 0:Interrupt disable 1:Interrupt enable
3	TSOIM	R/W	Time stamp counter overflow interrupt mask 0:Interrupt disable 1:Interrupt enable
2	BOIM	R/W	Bus off interrupt mask 0:Interrupt disable 1:Interrupt enable
1	EPIM	R/W	Error passive interrupt mask 0:Interrupt disable 1:Interrupt enable
0	WLIM	R/W	Warning level interrupt mask 0:Interrupt disable 1:Interrupt enable

The global interrupt mask register (CANGIM) controls whether to enable or disable a global interrupt correspondingly to each interrupt condition of the CANGIF register. When the bit in the CANGIF register is "0", the corresponding CAN global interrupt (INTCANGB) is disabled. When the bit in the CANGIF register is "1", the corresponding CAN global interrupt (INTCANGB) is enabled.

Reset operation clears all bits in the CANGIM register to "0", disabling global interrupts.

15.4.21 CANMBTIF (Mailbox Transmit Interrupt Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	-	MBTIF30	MBTIF29	MBTIF28	MBTIF27	MBTIF26	MBTIF25	MBTIF24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MBTIF23	MBTIF22	MBTIF21	MBTIF20	MBTIF19	MBTIF18	MBTIF17	MBTIF16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MBTIF15	MBTIF14	MBTIF13	MBTIF12	MBTIF11	MBTIF10	MBTIF9	MBTIF8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MBTIF7	MBTIF6	MBTIF5	MBTIF4	MBTIF3	MBTIF2	MBTIF1	MBTIF0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read : Read as "0". Write : Write as "0".
30-0	MBTIF30 to MBTIF0	R/W	Mailbox transmit interrupt flag (Each bit corresponds with mailboxes 30 to 0.) When the message in mailbox x has been successfully transmitted and the interrupt mask of the CANMBIM register is enabled (<MBIMx>="1"), the <MBTIFx> bit is set to "1" and the transmit completion interrupt (INTCANTX) becomes the "High" level. When CANMBIM<MBIMx> bit is "0", the <MBTIFx> bit is not set and INTCANTX stays at the "Low" level. Transmission completion is checked by reading the CANTA register. If even one bit in the CANMBTIF register is "1", INTCANTX is the "High" level. The <MBTIFx> bit is cleared by a write of "1" to the <MBTIFx> bit from the CPU. A write of "0" is invalid.

When the mailbox is set to receive, the corresponding bit in the CANMBTIF register is read as "0". When the mailbox is set to transmit, the corresponding bit in the CANMBRIF register is read as "0".

15.4.22 CANMBRIF (Mailbox Receive Interrupt Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	MBRIF31	MBRIF30	MBRIF29	MBRIF28	MBRIF27	MBRIF26	MBRIF25	MBRIF24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MBRIF23	MBRIF22	MBRIF21	MBRIF20	MBRIF19	MBRIF18	MBRIF17	MBRIF16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MBRIF15	MBRIF14	MBRIF13	MBRIF12	MBRIF11	MBRIF10	MBRIF9	MBRIF8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MBRIF7	MBRIF6	MBRIF5	MBRIF4	MBRIF3	MBRIF2	MBRIF1	MBRIF0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	MBRIF31 to MBRIF0	R/W	Mailbox receive interrupt flag (Each bit corresponds with mailboxes 31 to 0.) When mailbox x has successfully received the message and the interrupt mask of the CANMBIM register is enabled (<MBIMx> = "1"), the <MBRIFx> bit is set to "1" and the receive completion interrupt (INTRX) becomes the "High" level. When the <MBIMx> bit in the MBIM register is "0", the <MBRIFx> bit is not set and INTRX stays at the "Low" level. Receive completion is checked by reading the CANRMP register. If even one bit in the CANMBRIF register is "1", INTCANRX is the "High" level. The <MBRIFx> bit is cleared by a write of "1" to the <MBRIFx> bit from the CPU. A write of "0" is invalid.

15.4.23 CANMBIM (Mailbox Interrupt Mask Register)

	31	30	29	28	27	26	25	24
bit symbol	MBIM31	MBIM30	MBIM29	MBIM28	MBIM27	MBIM26	MBIM25	MBIM24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MBIM23	MBIM22	MBIM21	MBIM20	MBIM19	MBIM18	MBIM17	MBIM16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MBIM15	MBIM14	MBIM13	MBIM12	MBIM11	MBIM10	MBIM9	MBIM8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MBIM7	MBIM6	MBIM5	MBIM4	MBIM3	MBIM2	MBIM1	MBIM0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	MBIM31 to MBIM0	R/W	Mailbox interrupt mask 0:Interrupt disabled for corresponding mailbox 1:Interrupt enabled for corresponding mailbox

The settings in CANMBIM determine, for which mailbox the interrupt generation is enabled or disabled. If a bit in CANMBIM is "0", the interrupt generation for the corresponding mailbox is disabled and if it is "1", the interrupt generation is enabled. Reset value of CANMBIM is "0".

15.4.24 CANCDR (Change Data Request Register)

	31	30	29	28	27	26	25	24
bit symbol	-	CDR30	CDR29	CDR28	CDR27	CDR26	CDR25	CDR24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CDR23	CDR22	CDR21	CDR20	CDR19	CDR18	CDR17	CDR16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CDR15	CDR14	CDR13	CDR12	CDR11	CDR10	CDR9	CDR8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CDR7	CDR6	CDR5	CDR4	CDR3	CDR2	CDR1	CDR0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	−	R	Read : Read as "0". Write : Write as "0".
30-0	CDR30 to CDR0	R/W	Change data request (Each bit corresponds with mailboxes 30 to 0.) When the <CDRx> bit of transmit mailbox x is set to "1", the transmit request of this mailbox x is ignored. It means mailbox x for which the CANTRS<TRSx> bit and the <CDRx> bit are set will be excluded from the internal arbitration range and will not be transmitted if transmission has not started. After the <CDRx> bit is cleared to "0", mailbox x is back to be included in the internal arbitration range.

Note: Mailbox 31 is receive-only mailbox.

The change data request register CABCDR is effective when updating the data field of transmit mailbox x where auto acknowledgement of remote frames is enabled (CANMBnID<RFH>="1"). Mailbox x enabling automatic acknowledgement starts message transmission automatically responding to received remote frames and so may update the data field during message transmission (In such cases, updated data is output midway through transmission). The update of the data field can be avoided by setting the <CDRx> bit to "1" and temporarily suspending data transmission.

15.4.25 CANRFP (Remote Frame Pending Register)

	31	30	29	28	27	26	25	24
bit symbol	RFP31	RFP30	RFP29	RFP28	RFP27	RFP26	RFP25	RFP24
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RFP23	RFP22	RFP21	RFP20	RFP19	RFP18	RFP17	RFP16
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-0	RFP31 to RFP0	R/W	Remote frame pending (Each bit corresponds with mailboxes 31 to 0.) When mailbox x configured as receive mailbox receives a remote frame, the <RFPx> bit and CANRMP<RMPx> bit are set to "1". The <RFPx> bit can be cleared by a write of "1" to the CANRMP<RMPx> bit.

The CANRFP<RFPx> bit is set by the internal logic and can be cleared with a write of "1" to the CANRMP<RMPx> bit from the CPU. The <RMPx> bit is also cleared at the same time. A write of "0" to the <RMPx> bit and a write of "1" or "0" to the <RFPx> bit from the CPU are invalid.

Even when mailbox x with <RFPx>="1" is overwritten by data frame reception, the <RFPx> bit is cleared.

When the remote frame pending interrupt is enabled by setting the <RFPM> bit in the global interrupt mask register CANGIM to "1", the CAN global interrupt INTCANGB occurs.

15.4.26 CANCEC (Error 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	TEC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	REC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read : Read as "0". Write : Write as "0".
15-8	TEC[7:0]	R	8-bit transit error counter (After reset release)
		R/W	8-bit transit error counter (CANMCR<TSTERR>="1")
7-0	REC[7:0]	R	8-bit receive error counter (After reset release)
		R/W	8-bit receive error counter (CANMCR<TSTERR>="1")

The CAN controller contains two error counters : the receive error counter <REC> and the transmit error counter <TEC>. The value of both counters can be read from the CPU. A write access to the error counters is only possible in test error mode (The <TSTERR> bit in the CANMCR register is "1"). In the case of a write to the CANCEC register, the write data to the lower 8 bits <REC> is written also to the higher 8 bits (TEC).

The CAN error counters count up or down according to the CAN Specification 2.0B.

The <REC> is not increased after exceeding the error passive limit (128).When <REC>=128, after the correct reception of a message, the <REC> is set to a value between 119 and 127. After reaching the "bus off" status, the error counters are undefined.

If the status "bus off" is reached, the receive error counter is incremented after 11 consecutive recessive bits on the bus. If the counter reaches the count 128, the module changes automatically to the status error active. All internal flags are reset and the error counters will be cleared to "0". The configuration registers keep the programmed values. The values of the counters are undefined during "bus off" status.

When CAN enters configuration mode, the error counters will be cleared.

15.4.27 CANTSP (Time Stamp Counter Prescaler Register)

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

Bit	Bit Symbol	Type	Function
31-4	−	R	Read : Read as "0". Write : Write as "0".
3-0	TSP[3:0]	R/W	Time stamp counter prescaler Sets the value to be loaded to the prescaler for the 4-bit TSC.

To ensure that the value of the CANTSC will not change during the write cycle to the mailbox, a hold register is implemented. The value of the CANTSC will be copied to the hold register and then written to the mailbox from the hold register if a message has been received or transmitted successfully. The reception is successful for the receiver, if there is no error but the last one bit of End-of-frame. Transmission is successful for the transmitter if there is no error until the last bit of End-of-frame. (Refer to the CAN specification 2.0B).

15.4.28 CANTSC (Time Stamp 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	TSC							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TSC							
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	−	R	Read : Read as "0". Write : Write as "0".
15-0	TSC[15:0]	R/W	Time stamp counter Free-running 16-bit counter

Overflow of the CANTSC can be detected by the time stamp counter overflow interrupt flag <TSOIF> of the global interrupt flag register (CANGIF), and the time stamp counter overflow flag <TSO> of the global status register (CANGSR). Both flags can be cleared by writing "1" to <TSOIF> in the CANGIF register.

There is a 4-bit prescaler for the CANTSC. After power-up the time stamp counter is driven directly from the bit clock (<TSP[3:0]>="0"). The period T_{TSC} for the time stamp counter will be calculated with the following formula :

$$T_{TSC} = TBIT \times (TSP + 1)$$

15.5 Operation explain of each circuit

15.5.1 Mailbox

The mailboxes consist of a single port RAM (accessible from the internal CAN core and the CPU). The CPU controls the CAN controller by changing the settings of the mailboxes and control registers. The settings of the mailboxes and control registers are used for such processes as reception filtering, message transmission, and interrupt processing.

To start transmission, set the transmit request bit corresponding to the mailbox to transmit messages to. After the bit has been set, all transmission procedures and error processes (when errors occur) are executed without CPU involvement. When the mailbox is set to receive, the CPU reads the mailbox data using read instructions. The user can also set it so that an interrupt will be issued to the CPU every time a message has been successfully received or transmitted

In total, 32 mailboxes are provided, each of which consists of 8 byte data, 29 bit IDs, and several control bits. The mailboxes (except mailbox 31) can be set to either transmission or reception. Mailbox 31 is the receive-only mailbox. Mailbox 31 is designed so that it can receive different message ID groups using other receive masks than mailboxes 0 to 30.

Figure 15-2 shows the configuration of the mailboxes.

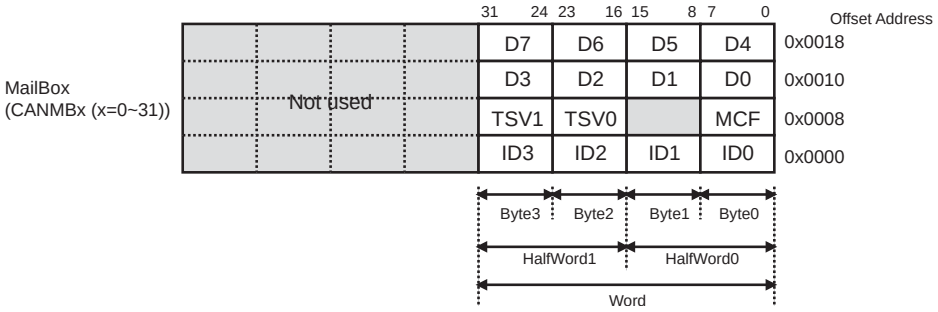


Figure 15-2 Configuration of Mailboxes

1. Message ID field (ID3 to ID0)
   - ID extension bit <IDE>
   - Global / local acceptance mask enable bit <GAME / LAME>
   - Remote frame handling bit <RFH>
   - 29-bit message ID <ID[28:0]>

2. Message control field (MCF)
   - Remote frame transmit request bit <RTR>
   - Data length of 4 bits <DLC[3:0]>

3. Time stamp value (TSV1, TSV0)

   Stores time stamp counter value during receiving / transmitting message. <TSV[15:0]>

4. Data field (D7 to D0)

   Data of 8 bytes <D7[7:0]> to <D0[7:0]>

15.5.2 Transmit Control Register

Transmission control consists of two registers. One is the transmission request set register CANTRS, and the other is the transmission request reset register CANTRR. Therefore it is possible to clear the transmission request without generating a conflict in the handling of the transmit mailboxes in the state-machine. This mechanism also prevents clearing the transmission request of a mailbox to which transmission is already in progress.

When a write of data and the ID to mailbox x configured as a transmit mailbox (CANMD<MDx>="0") is performed and access to mailbox x is enabled (CANMC<MCx>="1"), setting the CANTRS<TRSx> bit to "1" causes the messages in mailbox x to be transmitted.

If there is more than one mailbox configured as a transmit mailbox and more than one corresponding TRS bit is set, then the messages will be sent in the selected order. The order of transmission depends on the <MTOS> bit in the master control register CANMCR.

If the CANMCR<MTOS> bit is "0", the mailbox with the lower number has the higher priority. For example, if the mailboxes CANMB0, CANMB2, and CANMB5 are configured as transmit mailboxes and the corresponding CANTRS<TRSx> bits are set to "1", then the messages will be transmitted in the following order: CANMB0, CANMB2, and CANMB5. If a new transmission request is set for CANMB0 during processing of the CANMB2 message, then in the next internal arbitration-run, CANMB0 is selected for the next transmit message and transmission of the CANMB0 message starts after CANMB2 transmission is completed. This will also happen when an arbitration lost error occurs when the CANMB2 message is being transmitted. The CANMB0 message will be sent instead of the CANMB2 lost in arbitration.

If the CANMCR<MTOS> bit is "1", the mailbox with the highest priority ID among those mailboxes for which transmission is requested will be transmitted. In a transmission after an arbitration lost error occurred also, the message in the mailbox with the highest priority ID among those mailboxes for which transmission is requested at the time will be transmitted.

15.5.3 Receive Control Register

The ID of a received message is compared to the ID of the mailbox set as the receive mailbox. The comparison of the IDs depends on the <GAME> / <LAME> values of the global/local acceptance mask enable bit MBnID3 in the mailbox and the data held in the global/local acceptance mask registers CANGAM/CANLAM.

When a match is detected, the ID of the received message, the control bits, and data bytes are written in the matching mailbox. At the same time, when the corresponding receive message pending bit CANRMP<RMPx> is set to "1" and the mailbox interrupt is enabled (CANMBIM<MBIMx>="1"), the CAN receive completion interrupt INTCANRX occurs. After a match is detected, no further ID comparison takes place.

If the ID of the received message does not match with any of the mailboxes 0 to 30, the ID is compared to the ID of the receive-only mailbox 31. When a match is detected, the settings of the received message are written in receive-only mailbox 31.

If no match is detected, the received message will not be stored in the mailbox and no change occurs in the mailbox.

The <RMPx> bit must be cleared by the CPU after data is read. With the <RMPx> bit set to "1", if the next message to this mailbox x is received, the corresponding receive message lost bit <RMLx> is set to "1". In this case, mailbox x is overwritten with the new message.

Figure 15-3 shows timing when a receive message lost occurs.

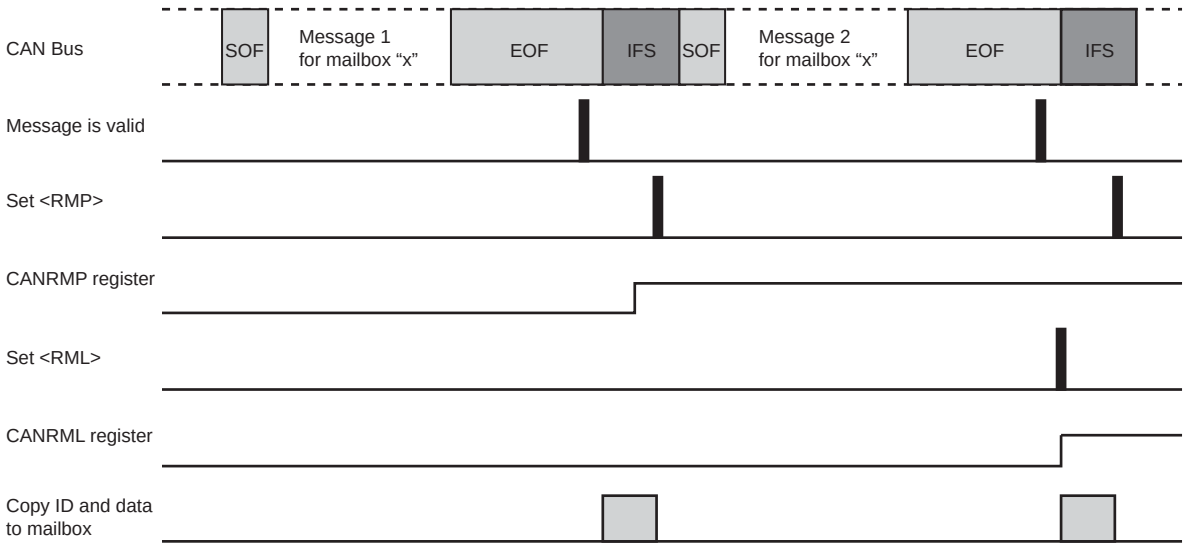


Figure 15-3 Timing when a Receive Message Lost Occurs

15.5.4 Remote Frame Control Register

After a remote frame is received, the remote frame ID is compared to the mailbox ID. The comparison of the IDs depends on the <GAME> / <LAME> values of the global / local acceptance mask enable bit CANMBxID in the mailbox and the data held in the global/local acceptance mask registers GAM/LAM.

After an ID match is detected, no further comparison takes place.

When the remote frame ID matches with the ID of transmit mailbox n where the remote frame handling bit CANMBxID<RFH> is set to "1", the CANTRS<TRSx> bit will be set to "1" so that the message is transmitted responding to the remote frame. For a transmit mailbox where the CANMBxID<RFH> bit is set to "0", the mailbox does not respond to the remote frame even when the ID matches.

If the ID matches with the ID of receive mailbox n, the received message is handled same as a data frame, and this sets the CANRMP<RMPx> bit and the CANRFP<RFPx> bit to "1".

When the remote frame ID matches with the ID of mailbox n where both the CANMBxID<RFH> bit and the <GAME> bit are set to "1", the ID of mailbox x is overwritten with the remote frame ID, and the mailbox will automatically respond to the remote frame using the ID (The <TRSx> bit is set to transmit a data frame). Therefore, when the global acceptance mask register CANGAM is used, one mailbox x may respond to multiple remote frame IDs depending on the mask value.

15.5.5 Receive Filtering

For mailboxes 0 to 30, the global acceptance mask register CANGAM will be used if the bit <GAME> in the mailbox is set. The receiving message will be stored in the first mailbox with a matching ID. Only if there is no matching ID in the mailboxes 0 to 30, the receiving message will be compared to the receive-only mailbox (mailbox 31). If the <LAME> bit in mailbox 31 is set, the local acceptance mask register CANLAM will be used.

Figure 15-4 shows receive filtering.

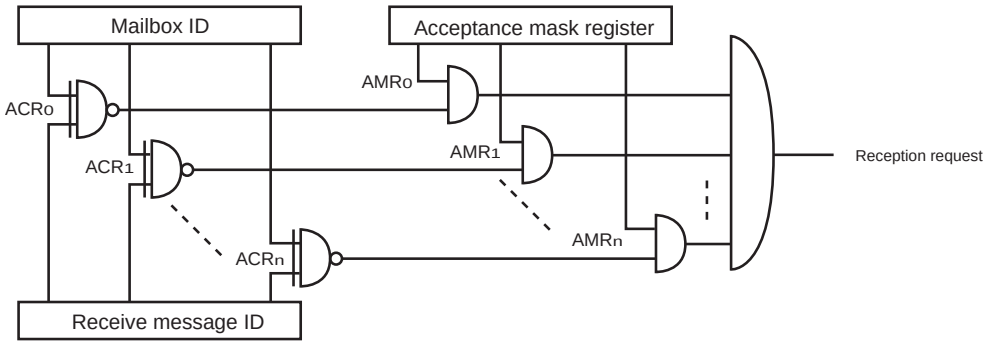


Figure 15-4 Receive filtering

15.5.6 Time Stamp Function

There is a free-running 16-bit time stamp counter (CANTSC) implemented in the CAN controller to show the time of message reception and transmission. The content of the CANTSC is written into the time stamp value (TSV) of the corresponding mailbox when a received message has been stored or a message has been transmitted.

The CANTSC is driven by the bit clock of the CAN bus line. When the operation mode of the CAN is in configuration mode or in sleep mode, the CANTSC will be stopped. After power-up reset, a write to the time stamp counter prescaler register (CANTSP) clears the CANTSC to "0". The CANTSC is readable and writable from the CPU both in configuration mode and normal operation mode.

Figure 15-5 shows the structure of the time stamp counter.

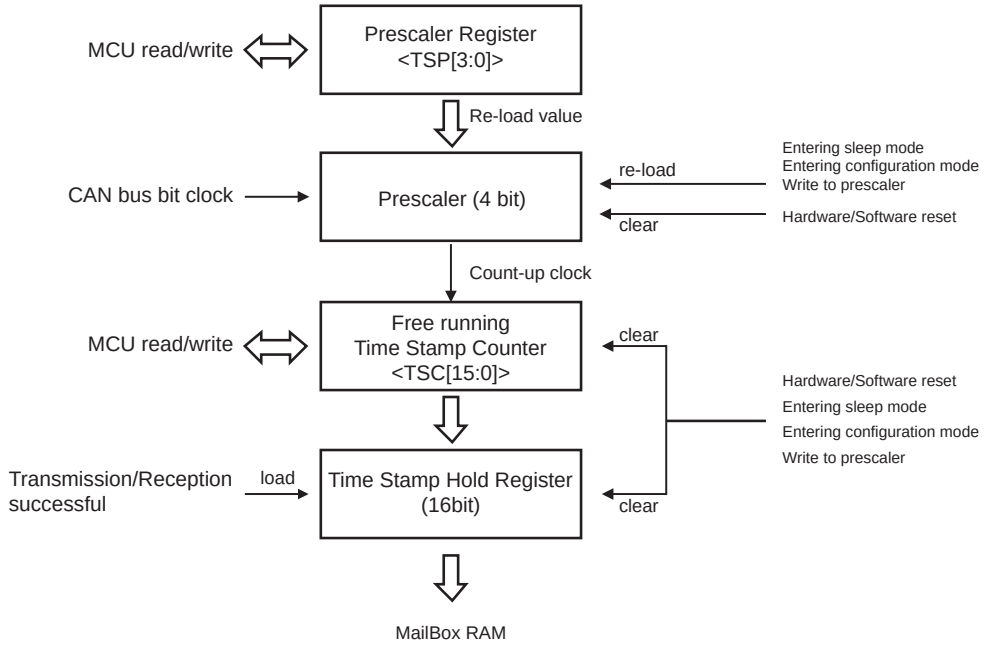


Figure 15-5 Timer Stamp Counter

The free running time stamp counter and the time stamp hold register will be cleared in the following cases:

- After reset (Power on reset or software reset)
- When the controller enters into configuration mode
- When the controller enters into sleep mode
- When a write access is performed to the CANTSP register.

15.5.7 Interrupt Control

The CAN controller has the following interrupt sources. And these interrupt sources are divided into three groups and each group has one interrupt output.

- CAN transmit completion interrupt (INTCANTX)

  It occurs at the completion of transmission
- CAN receive completion interrupt (INTCANRX)

  It occurs at the completion of reception
- CAN Global interrupt (INTCANGB)

  It occurs by eight sources other than those above.

Sources		Group
Transmit interrupt	:a message has been transmitted successfully.	INTCANTX
Receive interrupt	:a message has been received successfully.	INTCANRX
Warning level interrupt	: at least one of the two error counters is greater than or equal to 97.	INTCANGB
Error passive interrupt	:CAN enters the error passive mode.	
Bus off interrupt	:CAN enters the bus off mode.	
Time stamp overflow interrupt		
Transmit abort interrupt		
Receive message lost interrupt		
Walk-up interrupt	:after walk-up from sleep mode, this interrupt will be generated.	
Remote frame receive interrupt		

For mailbox interrupts, there are two interrupt output lines separated from global interrupts. These are the mailbox receive completion interrupt (INTCANRX) and the mailbox transmit completion interrupt (INTCANTX), which are dependent on mailbox settings.

There are two interrupt flag registers and one interrupt mask register. One interrupt flag register is for the mailbox receive interrupt flag register (CANMBRIF) and one for the mailbox transmit interrupt flag register (CANMBTIF). In addition, there is the mailbox interrupt mask register (CANMBIM) for setting whether to enable or disable each mailbox interrupt. The CANMBIM register is used both for transmit and receive mailboxes.

Figure 15-6 shows the block diagram of CAN interrupt signal.

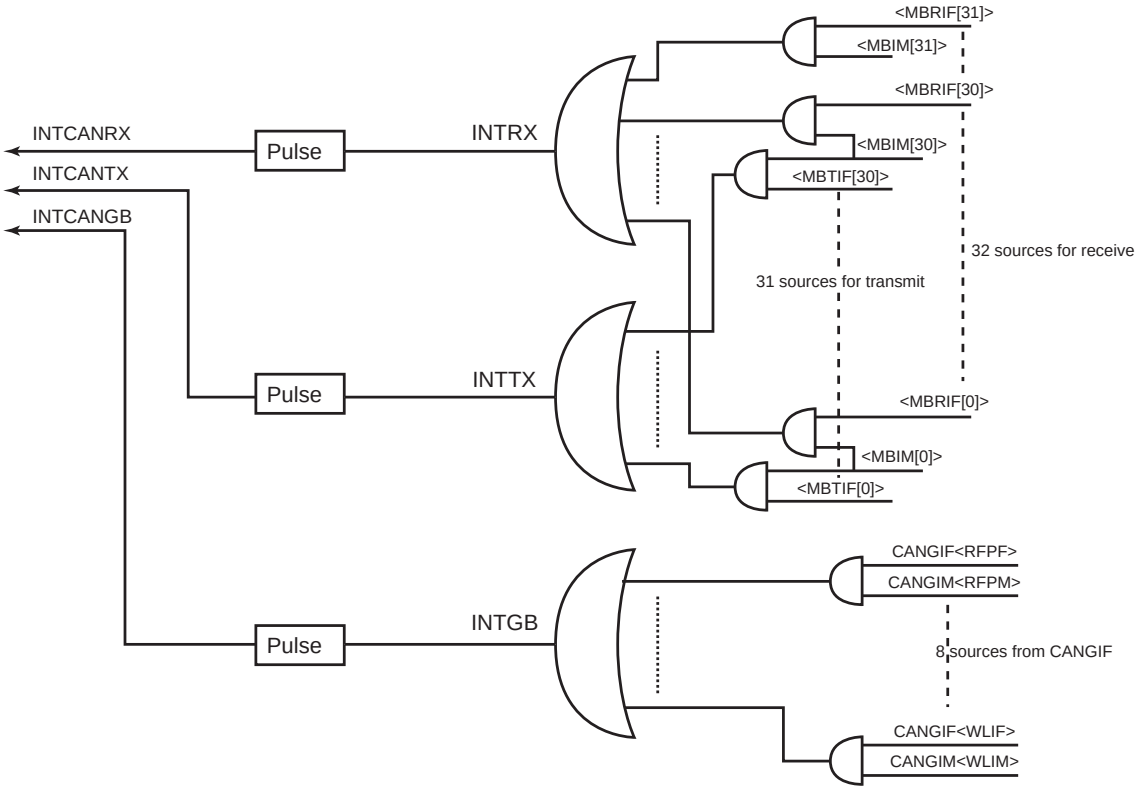


Figure 15-6 Block Diagram of CAN interrupt signals

The CAN receive completion interrupt signal INTRX is the OR of the signal for which the 32 sources issued by the mailbox receive interrupt flag register CANMBRIF that are ANDed with each bit of the mailbox interrupt mask register CANMBIM.

The CAN transmit completion interrupt signal INTTX is the OR of the signal for which the 31 sources issued by the mailbox transmit interrupt flag register CANMBTIF that are ANDed with each bit of the mailbox interrupt mask register CANMBIM.

The CAN global interrupt signal INTGB is the OR of the signal for which the 8 sources issued by the global interrupt flag register CANGIF that are ANDed with each bit of the global interrupt mask register CANGIM.

15.6 Operation Mode

15.6.1 Configuration Mode

The CAN controller needs initial setup before starting operation (setting of the bit configuration registers, CANBCR1 and CANBCR2). Writes to the CANBCR1 and CANBCR2 are possible only when the CAN controller is in configuration mode.

After reset, the CANMCR<CCR> and the CANGSR<CCE> are set to "1" and the configuration mode is set. A write of "0" to the <CCR> bit sets the CAN controller to normal operation mode. After leaving configuration mode, the <CCE> bit is cleared to "0" and the power-up sequence starts. The power-up sequence detects 11 consecutive recessive bits on the CAN bus line. After detection, the CAN controller is bus on and ready for operation.

A write of "1" to the <CCR> bit sets the CAN controller to enter configuration mode from normal operation mode. After the CAN controller has entered configuration mode, the <CCE> bit is set to "1".

Figure 15-7 shows the flowchart of the initial setup of the CAN controller.

When the CAN controller enters into configuration mode, the CAN error counter (CANCEC), the time stamp counter (CANTSC), and the time stamp hold registers will be cleared.

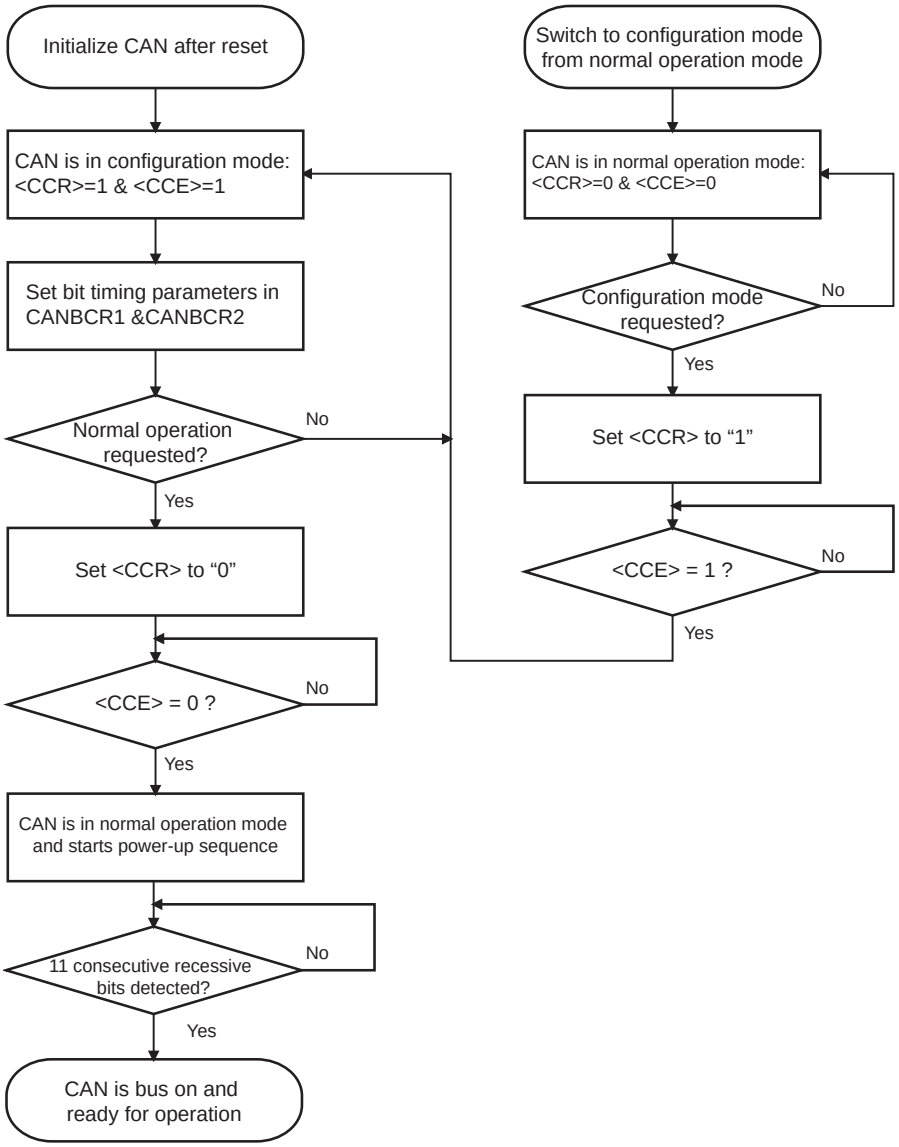


Figure 15-7 Flowchart of Initial Setup of CAN Controller

15.6.2 Sleep Mode

Sleep mode is requested by a write of "1" to the <SMR> bit in the CANMCR register. After the CAN controller has entered into sleep mode, the CANGSR<SMA> bit is set to "1".

The read value of the CANGSR register is 0xF040. This means that there is no message in the transmit buffer and sleep mode is active where the <SMA> bit is "1". Read values to all other registers deliver the value 0x0000. Write accesses to all registers, except the CANMCR register, will be denied.

The CAN controller cancels sleep mode (wakes up) and starts the power-up sequence if a write access to the CANMCR register is detected, or there is any bus activity detected on the CAN bus with the CANMCR <WUBA> bit set to "1". The CAN controller waits until detecting 11 consecutive recessive bits on the CANRX input terminal, after it goes into bus active state. The walk-up message is invalid.

In sleep mode, the CAN error counters and all transmission request set CANTRS<TRSx> bits and transmission request reset CANTRR<TRRx> bits are cleared. The <SMR> bit and the <SMA> bit are cleared after the CAN controller leaves sleep mode.

If sleep mode is requested while the CAN controller is transmitting a message (CANMCR<SMR>="1"), the CAN controller enters sleep mode after any of the following occurs:

- The message has been successfully transmitted.
- The message has been successfully transmitted after an arbitration lost error.
- The message has been successfully received after an arbitration lost error.

15.6.3 Suspend Mode

The suspend mode is requested by writing "1" to the CANMCR<SUR> bit. If the CAN bus line is not idle, the current message transmission/reception is completed before suspend mode is activated. After the CAN controller has entered suspend mode, the CANGSR<SUA> bit is set to "1".

In suspend mode, the CAN controller is not active on the CAN bus line. That means neither error frames nor acknowledgement will be sent. The error counters and the CANGSR<EP> bit will not be cleared either.

If suspend mode is requested during the bus off recovery sequence execution, the CAN controller enters suspend mode after the bus off recovery sequence is finished.

To restart the CAN controller, the <SUR> bit needs to be programmed to "0". After leaving the bus off state or the inactive state, the CAN controller restarts the bus off recovery sequence.

The CAN controller cancels suspend mode with a write of "0" to the <SUR> bit.

15.6.4 Test Loop Back Mode

In test loop back mode, the CAN controller can receive its own transmitted message and generates its own acknowledge bit. No other CAN node is necessary for the operation.

The test loop back mode can be enabled or disabled only when the CAN controller is in suspend mode. In test loop back mode, the CAN controller can transmit a message from a mailbox and receive it in another mailbox. The setup for mailboxes is the same as in normal operation mode.

15.6.5 Test Error Mode

In test error mode, writes to the CAN error counter register (CANCEC) are possible. The values of the lower 8 bits are concurrently written to both the transmit error counter (CANTEC) and the receive error counter (CANREC). The maximum value that can be written into the error counters is 255. The error counter value of 256 which forces the CAN controller into bus off mode cannot be written.

The test error mode can be enabled or disabled only when the CAN controller is in suspend mode.

Figure 15-8 shows the flowchart of the setup of test loop back mode and test error mode.

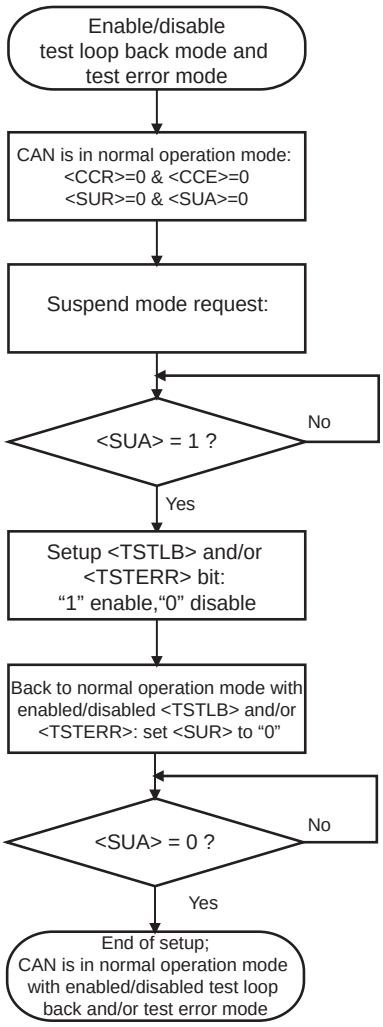


Figure 15-8 Flowchart of Setup of Test Loop Back Mode and Test Error Mode

15.7 Description of Operation

15.7.1 Receive Messages

Figure 15-9 shows an example flowchart of message reception using the CAN receive completion interrupt (INTCANRX).

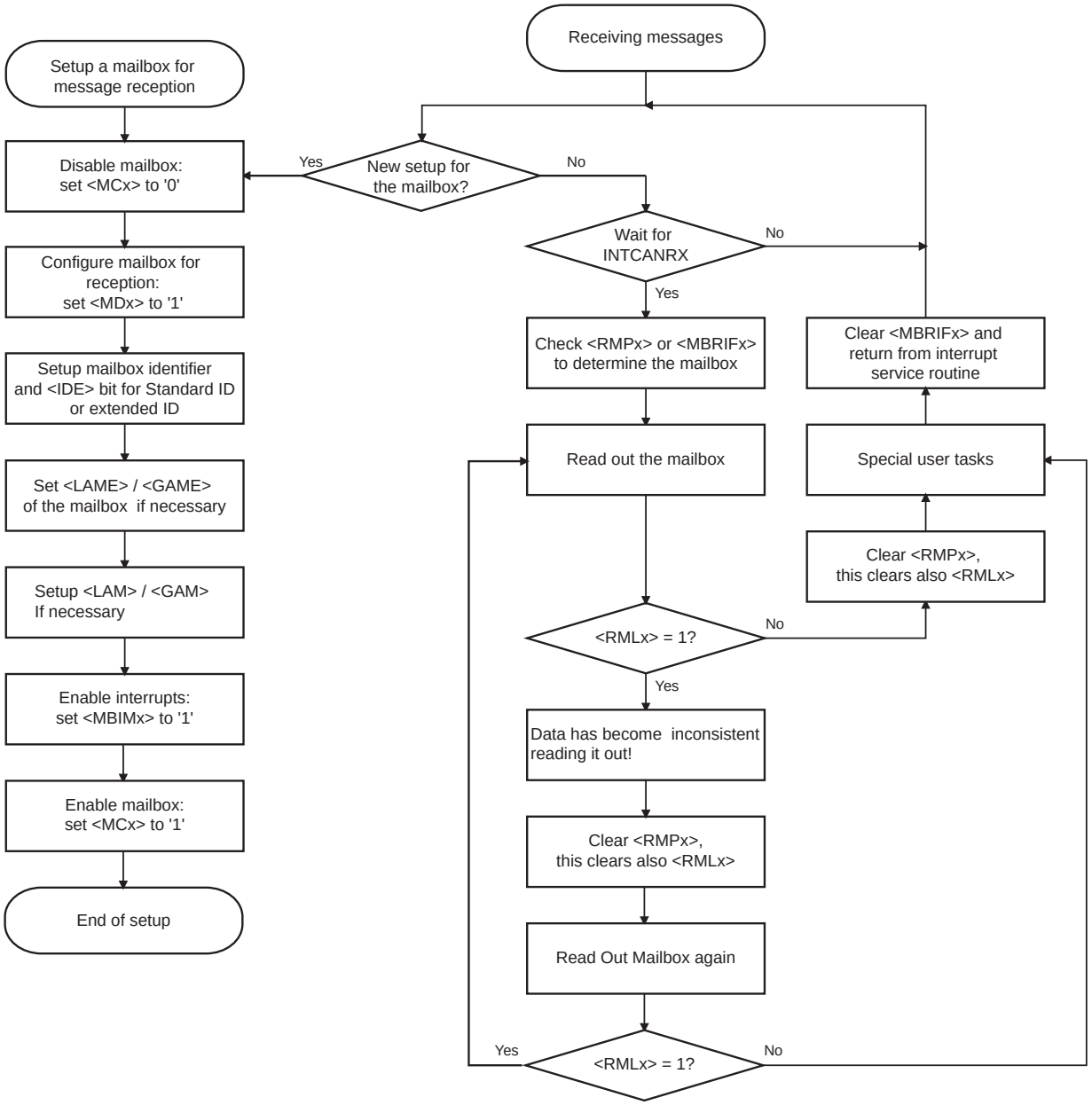


Figure 15-9 Flowchart of Message Reception

It is also possible to use polling instead of receive interrupts. In this case, the "waiting for INTCANRX" in above flowchart must be replaced by polling CANRMP. Further, enabling interrupts and clearing CANMBRIF must be removed from the flow.

15.7.2 Transmitting Message

Figure 15-10 shows an example flowchart of message transmission using the CAN transmit completion interrupt (INTCANTX).

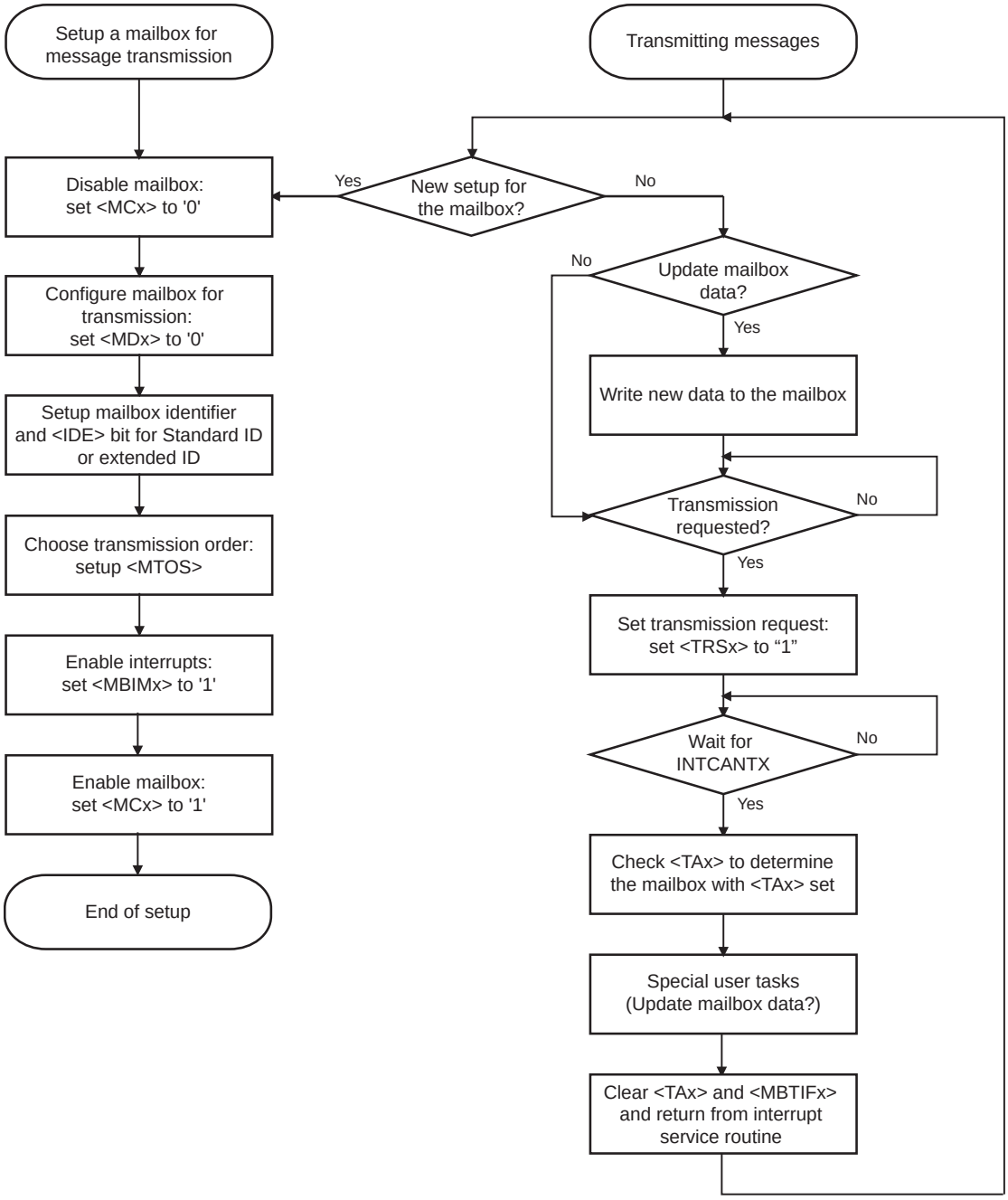


Figure 15-10 Flowchart of Message Transmission

It is also possible to use polling instead of transmit interrupts. In this case, the "waiting for INTCANTX" in above flowchart must be replaced by polling TA. Further, enabling interrupts and clearing CANMBTIF must be removed from the flow.

15.7.3 Remote Frame Handling

Figure 15-11 shows an example flowchart of remote frame handling by using the automatic reply feature. This feature is available when the <RFH> bit of the transmit mailbox is set to "1". To avoid data inconsistency when updating the mailbox data, the CANCDR register controls transmission during data update of the mailbox.

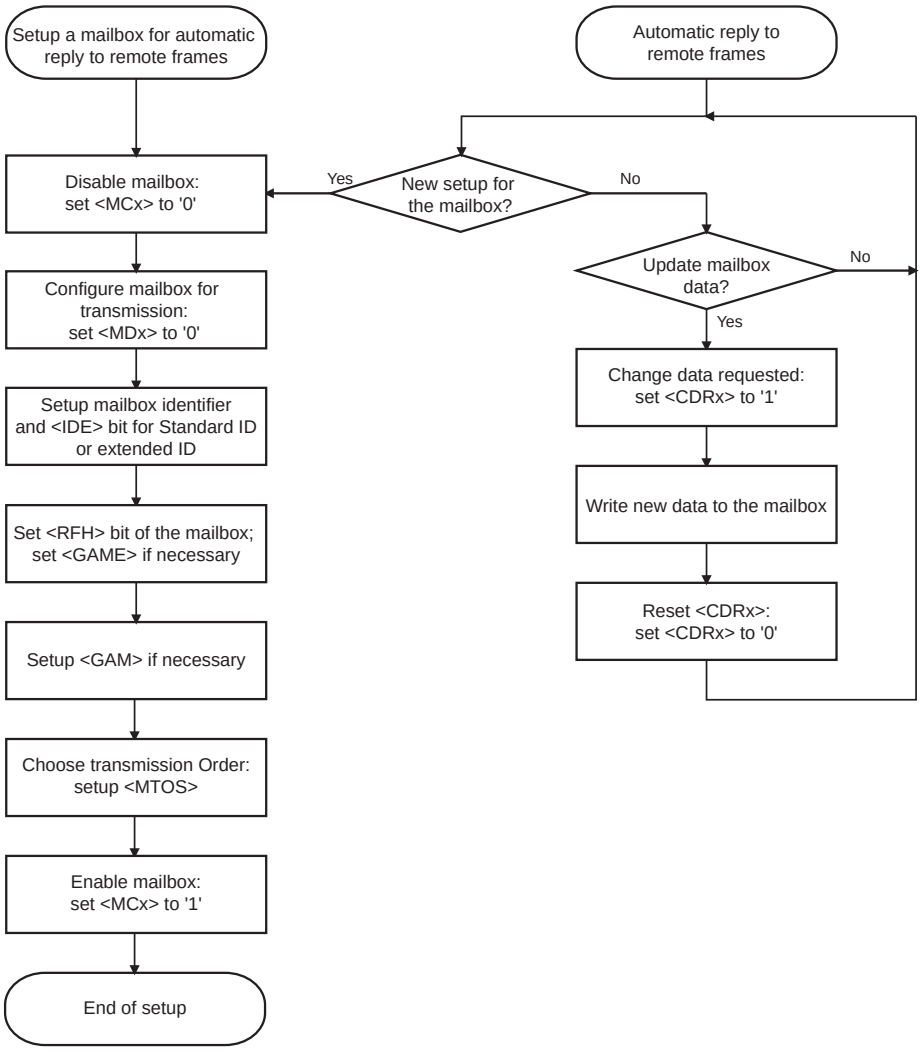


Figure 15-11 Flowchart of Remote Frame Handling Using the Automatic Reply Feature

15.8 Bit Configuration

The length of a bit is determined by the parameters TSEG1, TSEG2, and BRP. All controllers on the CAN bus must have the same baud rate and bit length. At different clock frequencies of the individual controllers, the baud rate has to be adjusted by the above-mentioned parameters. In the bit timing logic, the conversion of the parameters to the required bit timing is implemented. The configuration registers CANBCR1 and CANBCR2 contain the data about bit timing. Its definition corresponds to the CAN specification 2 (equivalent to Intel 82527).

Figure 15-12 shows CAN bit timing.

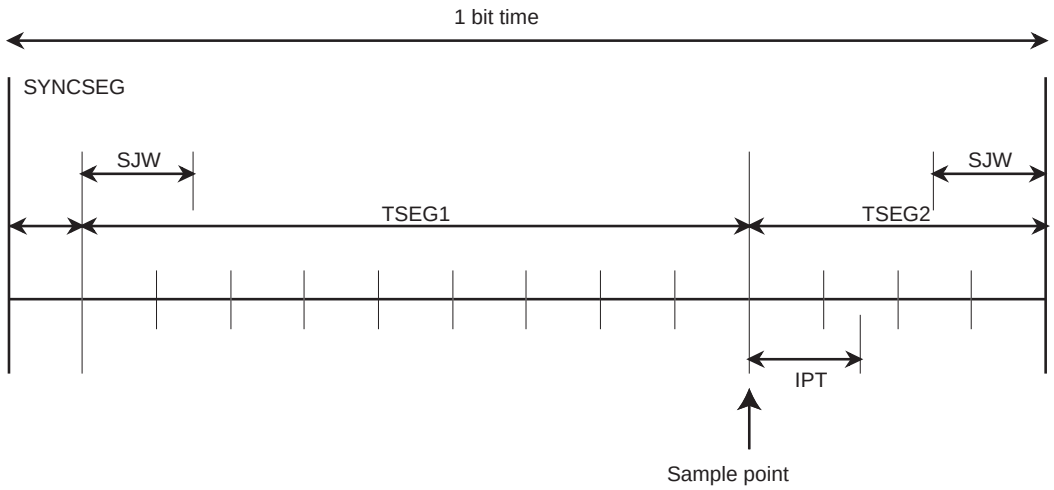


Figure 15-12 CAN Bit Timing

T_{SCL} (CAN system clock) is defined by :

$$T_{SCL} = \frac{<BRP[9:0]>+1}{f_{CANOSC}}$$

$1 \times T_{SCL} = 1 \times T_Q$ (T_Q : time quantum)

f_{CANOSC} is the clock for CAN baud rate generation. The clock obtained by dividing the system clock f_{SYS} by 4 is supplied as the clock for CAN baud rate generation. If f_{SYS} = 48MHz then f_{CANOSC} = 12MHz.

The synchronization segment SYNCSEG always has the length of one T_Q.

The baud rate is defined by :

$$BR = \frac{1}{((<TSEG1[3:0]>+1)+(<TSEG2[2:0]>+1)+1)\times T_{SCL}}$$

Note: <TSEG1[3:0]> and <TSEG2[2:0]> are values of the CANBCR2 register. It is not T_Q unit value.

Information processing time (IPT) is the time segment starting with the sample point reserved for processing of the sampled bit level. The information processing time is equal to three CAN system clock cycles.

<SJW[1:0]> indicates how much the time quantum (T_Q) value in bit length is allowed to be lengthened or shortened when resynchronizing. Values between "1" (<SJW[1:0]> = 00) and "4" (<SJW[1:0]> = 11) are adjustable. The bus line is sampled and synchronization is performed at each falling edge of the bus signal within a bit grid. For <SJW[1:0]>, set a value equal to or smaller than <TSEG2[2:0]>.

Setting the <SAM> bit enables the multiple sampling of the bus line. The level is determined by the result from the majority decision of three sampling values. Sampling is taken at the sample point and the previous last two CAN system clock points. When <BRP[9:0]> is smaller than 4, the sampling performed is always once regardless of the value set in the <SAM> bit.

Table 15-2 shows the restrictions when the baud rate is set.

Table 15-2 Restrictions when Setting the Baud Rate

<BRP[9:0]>	T_Q length (number of CAN clock cycles)	IPT length (number of CAN clock cycles)	Minimum TSEG2 length (T_Q unit)
0	1	3	3
1	2	3	2
> 1	<BRP[9:0]>+1	3	2

- Restrictions for TSEG1

  TSEG1 ≥ TSEG2 : The length of TSEG1 should be equal to or greater than the length of TSEG2.

- Restrictions for SJW

  SJW ≤ TSEG2 : For the synchronization jump width, set a value equal to or smaller than TSEG2.

- Restrictions for SAM

  The three-time sampling is not allowed under the condition that <BRP[9:0]> is smaller than 4. For the condition that <BRP[9:0]> < 4, a one-time sampling will always be performed regardless of the value of SAM.

Example : For 500 Kbit/s

A bit has a length of 2µs. If f_{CANOSC} = 12 MHz, the baud rate prescaler is set to "1". That means a bit for this data transmission rate has to be programmed with a length of $12T_Q$. According to the above formula, the values to be programmed always are smaller by one than the calculated values:

<BRP[9:0]> = 00_0000_0001

<TSEG1[3:0]> = 0110 (7 TQ)

<TSEG2[2:0]> = 011 (4 TQ)

In this case, the sample point is 8/12 = 66%.

Other combinations for TSEG1 / TSEG2 are possible; with TSEG2 = 3 the full range for SJW.

SJW should always be set to the highest value possible. SJW is not allowed to be greater than TSEG2.

The three-time sampling of the bus cannot be set because of the condition that <BRP[9:0]> is smaller than 4. Thus, SAM="0" should be set.

16. 12-Bit Analog-to-Digital Converter

16.1 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.

16.2 Block Diagram

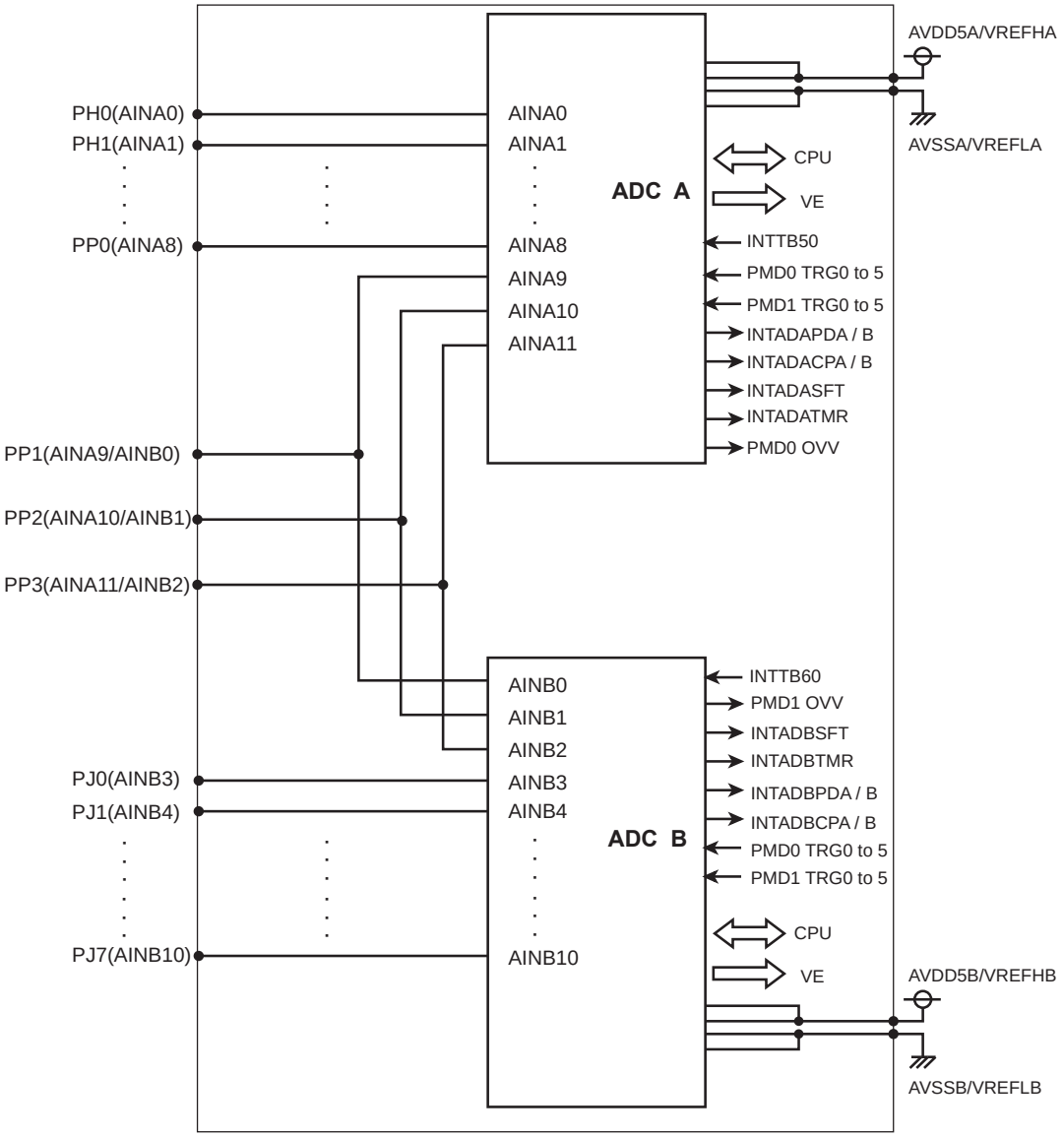


Figure 16-1 AD converters Block Diagram

16.3 List of Registers

The following table lists the control registers and their addresses:

For the base address, refer to "A list of peripheral function base addresses" in the chapter on "Memory Map."

Register Name		Address(Base+)
Clock Setting Register	ADxCLK	0x0000
Mode Setting Register 0	ADxMOD0	0x0004
Mode Setting Register 1	ADxMOD1	0x0008
Mode Setting Register 2	ADxMOD2	0x000C
Monitoring Setting Register 0	ADxCMPCR0	0x0010
Monitoring Setting Register 1	ADxCMPCR1	0x0014
Conversion Result Compare Register 0	ADxCMP0	0x0018
Conversion Result Compare Register 1	ADxCMP1	0x001C
Conversion Result Register 0	ADxREG0	0x0020
Conversion Result Register 1	ADxREG1	0x0024
Conversion Result Register 2	ADxREG2	0x0028
Conversion Result Register 3	ADxREG3	0x002C
Conversion Result Register 4	ADxREG4	0x0030
Conversion Result Register 5	ADxREG5	0x0034
Conversion Result Register 6	ADxREG6	0x0038
Conversion Result Register 7	ADxREG7	0x003C
Conversion Result Register 8	ADxREG8	0x0040
Conversion Result Register 9	ADxREG9	0x0044
Conversion Result Register 10	ADxREG10	0x0048
Conversion Result Register 11	ADxREG11	0x004C
PMD Trigger Program Number Select Register 0	ADxPSEL0	0x0050
PMD Trigger Program Number Select Register 1	ADxPSEL1	0x0054
PMD Trigger Program Number Select Register 2	ADxPSEL2	0x0058
PMD Trigger Program Number Select Register 3	ADxPSEL3	0x005C
PMD Trigger Program Number Select Register 4	ADxPSEL4	0x0060
PMD Trigger Program Number Select Register 5	ADxPSEL5	0x0064
PMD Trigger Program Number Select Register 6	ADxPSEL6	0x0068
PMD Trigger Program Number Select Register 7	ADxPSEL7	0x006C
PMD Trigger Program Number Select Register 8	ADxPSEL8	0x0070
PMD Trigger Program Number Select Register 9	ADxPSEL9	0x0074
PMD Trigger Program Number Select Register 10	ADxPSEL10	0x0078
PMD Trigger Program Number Select Register 11	ADxPSEL11	0x007C
PMD Trigger Interrupt Select Register 0	ADxPINTS0	0x0080
PMD Trigger Interrupt Select Register 1	ADxPINTS1	0x0084
PMD Trigger Interrupt Select Register 2	ADxPINTS2	0x0088
PMD Trigger Interrupt Select Register 3	ADxPINTS3	0x008C
PMD Trigger Interrupt Select Register 4	ADxPINTS4	0x0090
PMD Trigger Interrupt Select Register 5	ADxPINTS5	0x0094
PMD Trigger Program Register 0	ADxPSET0	0x0098
PMD Trigger Program Register 1	ADxPSET1	0x009C
PMD Trigger Program Register 2	ADxPSET2	0x00A0
PMD Trigger Program Register 3	ADxPSET3	0x00A4
PMD Trigger Program Register 4	ADxPSET4	0x00A8
PMD Trigger Program Register 5	ADxPSET5	0x00AC

Register Name		Address(Base+)
Timer Trigger Program Registers 0 to 3	ADxTSET03	0x00B0
Timer Trigger Program Registers 4 to 7	ADxTSET47	0x00B4
Timer Trigger Program Registers 8 to 11	ADxTSET811	0x00B8
Software Trigger Program Registers 0 to 3	ADxSSET03	0x00BC
Software Trigger Program Registers 4 to 7	ADxSSET47	0x00C0
Software Trigger Program Registers 8 to 11	ADxSSET811	0x00C4
Constant Conversion Program Registers0 to 3	ADxASET03	0x00C8
Constant Conversion Program Registers 4 to 7	ADxASET47	0x00CC
Constant Conversion Program Registers 8 to 11	ADxASET811	0x00D0
Mode Setting Register 3	ADxMOD3	0x00D4
Mode Setting Register 4	ADxMOD4	0x00D8
Mode Setting Register 5	ADxMOD5	0x00DC

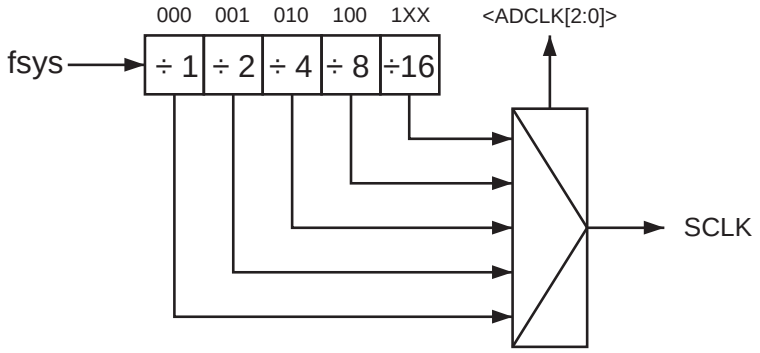
16.4 Register Descriptions

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

16.4.1 ADxCLK (Clock Setting Register)

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

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as "0".
6-3	EXAZ[3:0]	R/W	AIN sampling period is extended.. 0000: ([SCLK period] × 34) × 1 0001: ([SCLK period] × 34) × 2 0010: ([SCLK period] × 34) × 3 0011: ([SCLK period] × 34) × 4 0101: ([SCLK period] × 34) × 16 0111: ([SCLK period] × 34) × 64 1000: ([SCLK period] × 34) × 128 1001: ([SCLK period] × 34) × 256 1010: ([SCLK period] × 34) × 512 1011: ([SCLK period] × 34) × 1024 Other than those above: Reserved
2-0	VADCLK[2:0]	R/W	AD prescaler output (SCLK) select 000: fsys (Note1) 001: fsys/2 010: fsys/4 011: fsys/8 1xx: fsys/16



Note 1: Frequency of SCLK can be used up to 120MHz.

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.

16.4.2 ADxMOD0 (Mode Setting Register 0)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	DACON	R/W	DAC control 0: OFF 1: ON Setting <DACON> to "1", when using the ADC.
0	ADSS	W	Software triggered conversion 0: Don't care 1: Start Setting <ADSS> to "1" starts AD conversion (software triggered conversion). Receiving trigger signal from PMD or TMRB(interrupt) starts AD conversion also. For detail setting, please read the chapter about PMD and TMRB.

Note: To start the AD conversion, ADxMOD1<ADEN> is set to "1" after ADxMOD0<DACON> is set to "1". Then, perform software triggered conversion start or constant AD conversion control enable. It needs a stable time for 3μs after ADxMOD1<DACON> is set to "1".

16.4.3 ADxMOD1 (Mode Setting Register 1)

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

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as "0".
7	ADEN	R/W	AD conversion control 0: Disable 1: Enable Setting <ADEN> to "1", when using the ADC.
6-1	−	R	Read as "0".
0	ADAS	R/W	Constant AD conversion control 0: Disable 1: Enable After setting <ADEN> to "1", setting <ADAS> to "1" starts AD conversion and repeat conversion.

16.4.4 ADxMOD2 (Mode Setting Register 2)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1	ADSFN	R	Software conversion busy flag 0: Conversion completed 1: Conversion in progress The <ADSFN> is a software AD conversion busy flag. After <ADSS> was set to "1", when AD conversion is actually started, <ADSFN> is set to "1". When all AD conversion by softwear trigger is completed, <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".

16.4.5 ADxMOD3 (Mode Setting Register 3)

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

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15	−	R/W	Write as "0".
14	PBSEL	R/W	Write as "0".
13	−	R	Read as "0".
12	BINMOD	R/W	Write as "0".
11-10	BITS[1:0]	R/W	Write as "00".
9	−	R/W	Write as "0".
8	RCUT	R/W	Reduce power consumption 0: Normal operation 1: Iref cut 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".
7-5	BIAS[2:0]	R/W	Write as "000".
4-0	−	R	Read as "0".

16.4.6 ADxMOD4 (Mode Setting 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	-	-	-	-	PHASEC			
After reset	0	0	0	0	0	0	1	0
	7	6	5	4	3	2	1	0
bit symbol	PHASEB				-	PHASEA		
After reset	0	0	0	1	0	0	1	0

Bit	Bit Symbol	Type	Function
31-12	−	R	Read as "0".
11-8	PHASEC[3:0]	R/W	Write as "0001".
7-4	PHASEB[3:0]	R/W	Write as "0001".
3	−	R	Read as "0".
2-0	PHASEA[2:0]	R/W	Write as "001".

16.4.7 ADxMOD5 (Mode Setting 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	PHASEF			PHASEE				
After reset	0	1	1	0	0	1	1	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PHASED				
After reset	0	0	0	0	1	0	0	1

Bit	Bit Symbol	Type	Function
31-16	−	R	Read as "0".
15-13	PHASEF[2:0]	R/W	Write as "011".
12-8	PHASEE[4:0]	R/W	Write as "00110".
7-5	−	R	Read as "0".
4-0	PHASED[4:0]	R/W	Write as "01101".

16.4.8 ADxCMPCR0(Monitoring Setting Register 0)

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

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT0			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT0[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons ・ ・ 15: After 16 comparisons
7	CMP0EN	R/W	Monitoring function 0 enable 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	Select AD conversion result register to be compared with ADxCMP0. 0000: ADxREG0 0100: ADxREG4 1000: ADxREG8 0001: ADxREG1 0101: ADxREG5 1001: ADxREG9 0010: ADxREG2 0110: ADxREG6 1010: ADxREG10 0011: ADxREG3 0111: ADxREG7 1011: ADxREG11

16.4.9 ADxCMPCR1(Monitoring Setting Register 1)

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

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT1			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-12	-	R	Read as "0".
11-8	CMPCNT1[3:0]	R/W	Comparison count for determining the result 0: After every comparison 1: After two comparisons ・ ・ 15: After 16 comparisons
7	CMP1EN	R/W	Monitoring function 1 enable 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	Select AD conversion result register to be compared with ADxCMP1. 0000: ADxREG0 / 0100: ADxREG4 / 1000: ADxREG8 0001: ADxREG1 / 0101: ADxREG5 / 1001: ADxREG9 0010: ADxREG2 / 0110: ADxREG6 / 1010: ADxREG10 0011: ADxREG3 / 0111: ADxREG7 / 1011: ADxREG11

16.4.10 ADxCMP0(Conversion Result Compare Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP0				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP0[11:0]	R/W	The value to be compared with an AD conversion result. Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

16.4.11 ADxCMP1(Conversion Result Compare Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP1				-	-	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	AD0CMP1[11:0]	R/W	The value to be compared with an AD conversion result. Specify the value to be compared with an AD conversion result.
3-0	-	R	Read as "0".

16.4.12 ADxREG0(Conversion Result Register 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR0							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR0				-	-	OVR0	ADR0RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR0[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR0	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG0 is read and is cleared when the low-order byte of ADxREG0 is read.
0	ADR0RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR0RF> is a flag that is set when an AD conversion result is stored in the ADxREG0 register and is cleared when the low-order byte of ADxREG0 is read.

16.4.13 ADxREG1(Conversion Result Register 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR1							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR1				-	-	OVR1	ADR1RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR1[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR1	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG1 is read and is cleared when the low-order byte of ADxREG1 is read.
0	ADR1RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR1RF> is a flag that is set when an AD conversion result is stored in the ADxREG1 register and is cleared when the low-order byte of ADxREG1 is read.

16.4.14 ADxREG2(Conversion Result Register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR2							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR2				-	-	OVR2	ADR2RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR2[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR2	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG2 is read and is cleared when the low-order byte of ADxREG2 is read.
0	ADR2RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR2RF> is a flag that is set when an AD conversion result is stored in the ADxREG2 register and is cleared when the low-order byte of ADxREG2 is read.

16.4.15 ADxREG3(Conversion Result Register 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR3							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR3				-	-	OVR3	ADR3RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR3[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR3	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG3 is read and is cleared when the low-order byte of ADxREG3 is read.
0	ADR3RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR3RF> is a flag that is set when an AD conversion result is stored in the ADxREG3 register and is cleared when the low-order byte of ADxREG3 is read.

16.4.16 ADxREG4(Conversion Result Register 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR4							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR4				-	-	OVR4	ADR4RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR4[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR4	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG4 is read and is cleared when the low-order byte of ADxREG4 is read.
0	ADR4RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR4RF> is a flag that is set when an AD conversion result is stored in the ADxREG4 register and is cleared when the low-order byte of ADxREG4 is read.

16.4.17 ADxREG5(Conversion Result Register 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR5							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR5				-	-	OVR5	ADR5RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR5[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR5	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG5 is read and is cleared when the low-order byte of ADxREG5 is read.
0	ADR5RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR5RF> is a flag that is set when an AD conversion result is stored in the ADxREG5 register and is cleared when the low-order byte of ADxREG5 is read.

16.4.18 ADxREG6(Conversion Result Register 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR6							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR6				-	-	OVR6	ADR6RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR6[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR6	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG6 is read and is cleared when the low-order byte of ADxREG6 is read.
0	ADR6RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR6RF> is a flag that is set when an AD conversion result is stored in the ADxREG6 register and is cleared when the low-order byte of ADxREG6 is read.

16.4.19 ADxREG7(Conversion Result Register 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR7							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR7				-	-	OVR7	ADR7RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR7[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR7	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG7 is read and is cleared when the low-order byte of ADxREG7 is read.
0	ADR7RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR7RF> is a flag that is set when an AD conversion result is stored in the ADxREG7 register and is cleared when the low-order byte of ADxREG7 is read.

16.4.20 ADxREG8(Conversion Result Register 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR8							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR8				-	-	OVR8	ADR8RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR8[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR8	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG8 is read and is cleared when the low-order byte of ADxREG8 is read.
0	ADR8RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR8RF> is a flag that is set when an AD conversion result is stored in the ADxREG8 register and is cleared when the low-order byte of ADxREG8 is read.

16.4.21 ADxREG9(Conversion Result Register 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR9							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR9				-	-	OVR9	ADR9RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR9[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR9	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG9 is read and is cleared when the low-order byte of ADxREG9 is read.
0	ADR9RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR9RF> is a flag that is set when an AD conversion result is stored in the ADxREG9 register and is cleared when the low-order byte of ADxREG9 is read.

16.4.22 ADxREG10(Conversion Result Register 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR10							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR10	ADR10RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR10[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR10	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG10 is read and is cleared when the low-order byte of ADxREG10 is read.
0	ADR10RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR10RF> is a flag that is set when an AD conversion result is stored in the ADxREG10 register and is cleared when the low-order byte of ADxREG10 is read.

16.4.23 ADxREG11(Conversion Result Register 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR11							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR11				-	-	OVR11	ADR11RF
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-16	-	R	Read as "0".
15-4	ADR11[11:0]	R	The value of an AD conversion result
3-2	-	R	Read as "0".
1	OVR11	R	OverRun flag 0:No overrun occurred 1:Overrun occurred This flag is set when a new AD conversion result is stored before the value of ADxREG11 is read and is cleared when the low-order byte of ADxREG11 is read.
0	ADR11RF	R	AD conversion result store flag 0:No result stored 1:Result stored <ADR11RF> is a flag that is set when an AD conversion result is stored in the ADxREG11 register and is cleared when the low-order byte of ADxREG11 is read.

16.4.24 PMD Trigger Program Registers

AD conversion can be started by a trigger from the PMD (programmable motor driver).

The PMD trigger program registers are used to specify the program to be started by each of twelve triggers generated by the PMD, to select the interrupt to be generated upon completion of the program and to select the AIN input to be used.

The PMD trigger program registers include three types of registers.

- PMD Trigger Program Number Select Register (ADxPSEL0 to ADxPSEL11)

  The PMD Trigger Program Number Select Register (ADxPSELn) specifies the program to be started by each of twelve AD conversion start signals corresponding to twelve triggers(PMD0TRG0 to 5 , PMD1TRG0 to 5) generated by the PMD. Programs 0 to 5 are available.

  "ADxPSEL0 to ADxPSEL5" corresponds to "PMD0TRG0 to 5". "ADxPSEL6 to ADxPSEL11" corresponds to "PMD1TRG0 to 5".
- PMD Trigger Interrupt Select Register (ADxPINTS0 to ADxPINTS5)

  The PMD Trigger Interrupt Select Registers (ADxPINTS0 to ADxPINTS5) select the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

  ADxPINTS0 corresponds to program 0, and it exists to ADxPINT5 (program 5).
- PMD Trigger Program Register (ADxPSET0 to ADxPSET5)

  The PMD Trigger Program Setting Registers (ADxPSET0 to ADxPSET5) specify the settings for each of programs 0 to 5. Each PMD Trigger Program Register is comprised of four registers for specifying the AIN input to be converted. The conversion results corresponding to the ADxPSETn0 to ADxPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADxREG0 to ADxREG3).

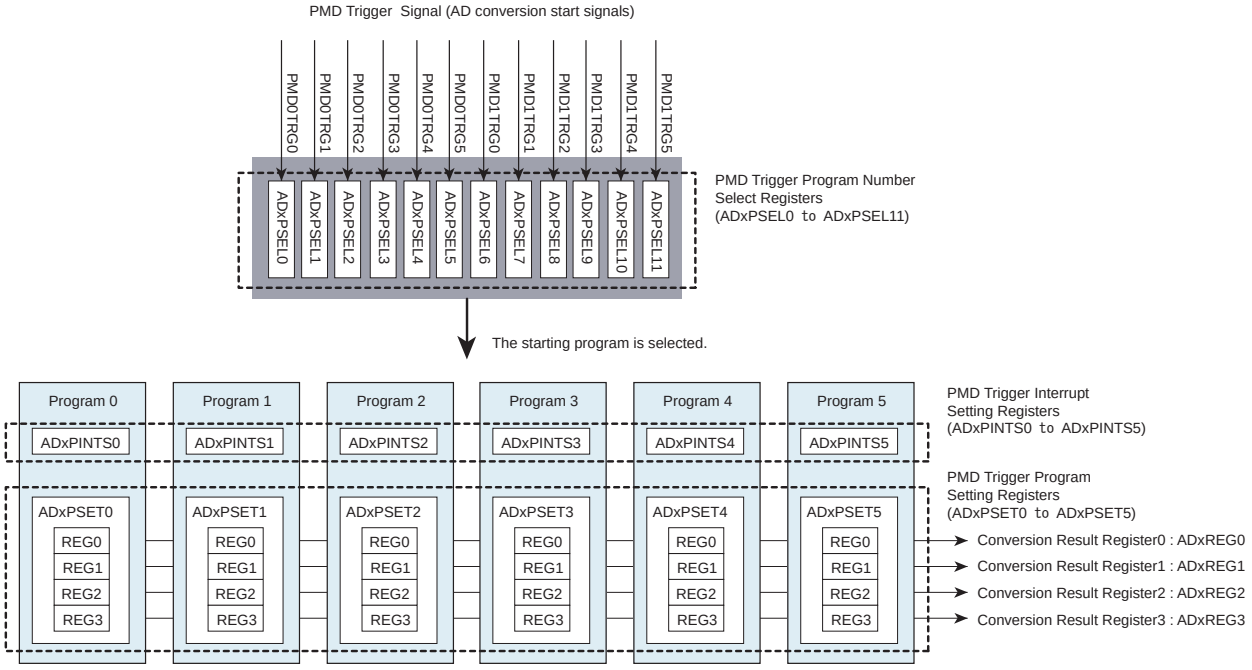


Figure 16-2 PMD Trigger Program Registers

16.4.24.1 ADxPSEL0 to ADxPSEL11(PMD Trigger Program Number Select Register 0 to 11)

ADxPSEL0:PMD Trigger Program Number Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS0	-	-	-	-	PMDS0		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS0	R/W	PMD0TRG0 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS0[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL1:PMD Trigger Program Number Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS1	-	-	-	-	PMDS1		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS1	R/W	PMD0TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS1[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL2:PMD Trigger Program Number Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS2	-	-	-	-	PMDS2		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS2	R/W	PMD0TRG2 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS2[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL3:PMD Trigger Program Number Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS3	-	-	-	-	PMDS3		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS3	R/W	PMD0TRG3 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS3[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL4:PMD Trigger Program Number Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS4	-	-	-	-	PMDS4		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS4	R/W	PMD0TRG4 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS4[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL5:PMD Trigger Program Number Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS5	-	-	-	-	PMDS5		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS5	R/W	PMD0TRG5 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS5[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL6:PMD Trigger Program Number Select Register 6

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS6	-	-	-	-	PMDS6		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS6	R/W	PMD1TRG0 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS6[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL7:PMD Trigger Program Number Select Register 7

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS7	-	-	-	-	PMDS7		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS7	R/W	PMD1TRG1 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS7[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL8:PMD Trigger Program Number Select Register 8

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS8	-	-	-	-	PMDS8		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS8	R/W	PMD1TRG2 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS8[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL9:PMD Trigger Program Number Select Register 9

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS9	-	-	-	-	PMDS9		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS9	R/W	PMD1TRG3 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS9[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL10:PMD Trigger Program Number Select Register 10

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS10	-	-	-	-	PMDS10		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS10	R/W	PMD1TRG4 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS10[2:0]	R/W	Program number select (Refer to Table 16-1)

ADxPSEL11:PMD Trigger Program Number Select Register 11

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS11	-	-	-	-	PMDS11		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as "0".
7	PENS11	R/W	PMD1TRG5 trigger control 0:Disable 1:Enable
6-3	-	R	Read as "0".
2-0	PMDS11[2:0]	R/W	Program number select (Refer to Table 16-1)

Table 16-1 Program number select

<PMDS0[2:0]>~ <PMDS11[2:0]>	
000	Program0
001	Program1
010	Program2
011	Program3
100	Program4
101	Program5
110	reserved
111	reserved

16.4.24.2 ADxPINTS0 to 5(PMD Trigger Interrupt Select Register 0 to 5)

ADxPINTS0:PMD Trigger Interrupt Select Register 0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL0	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL0[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 0.

ADxPINTS1:PMD Trigger Interrupt Select Register 1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL1	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL1[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 1.

ADxPINTS2:PMD Trigger Interrupt Select Register 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL2	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL2[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 2.

ADxPINTS3:PMD Trigger Interrupt Select Register 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL3	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL3[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 3.

ADxPINTS4:PMD Trigger Interrupt Select Register 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL4	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL4[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 4.

ADxPINTS5:PMD Trigger Interrupt Select Register 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL5	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-2	-	R	Read as "0".
1-0	INTSEL5[1:0]	R/W	Interrupt select 00:No interrupt output 01:INTADxPDA 10:INTADxPDB 11: No interrupt output The starting interrupt is selected for program 5.

16.4.24.3 ADxPSET0 to 5(PMD Trigger Program Register 0 to 5)

Each ADxPSETn (n=0 to 5:Program number) is composed of four sets that assume <AINSPnm [4:0]>, <UVWISnm[1:0]>, and <ENSPnm> in a couple. (m=0 to 3)

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.

ADxREGm ADxPSETn	m=0	m=1	m=2	m=3
n=0	<ENSP00> <UVWIS00> <AINSP00>	<ENSP01> <UVWIS01> <AINSP01>	<ENSP02> <UVWIS02> <AINSP02>	<ENSP03> <UVWIS03> <AINSP03>
n=1	<ENSP10> <UVWIS10> <AINSP10>	<ENSP11> <UVWIS11> <AINSP11>	<ENSP12> <UVWIS12> <AINSP12>	<ENSP13> <UVWIS13> <AINSP13>
n=2	<ENSP20> <UVWIS20> <AINSP20>	<ENSP21> <UVWIS21> <AINSP21>	<ENSP22> <UVWIS22> <AINSP22>	<ENSP23> <UVWIS23> <AINSP23>
n=3	<ENSP30> <UVWIS30> <AINSP30>	<ENSP31> <UVWIS31> <AINSP31>	<ENSP32> <UVWIS32> <AINSP32>	<ENSP33> <UVWIS33> <AINSP33>
n=4	<ENSP40> <UVWIS40> <AINSP40>	<ENSP41> <UVWIS41> <AINSP41>	<ENSP42> <UVWIS42> <AINSP42>	<ENSP43> <UVWIS43> <AINSP43>
n=5	<ENSP50> <UVWIS50> <AINSP50>	<ENSP51> <UVWIS51> <AINSP51>	<ENSP52> <UVWIS52> <AINSP52>	<ENSP53> <UVWIS53> <AINSP53>

Table 16-2 Select the AIN pin

<AINSP00 [4:0]> to <AINSP53 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 to 1_1111	:Reserved	

ADxPSET0:PMD Trigger Program Register 0

	31	30	29	28	27	26	25	24
bit symbol	ENSP03	UVWIS03		AINSP03				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP02	UVWIS02		AINSP02				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP01	UVWIS01		AINSP01				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP00	UVWIS00		AINSP00				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP03	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS03[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP03[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSP02	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS02[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP02[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSP01	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS01[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP01[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSP00	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS00[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP00[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET1:PMD Trigger Program Register 1

	31	30	29	28	27	26	25	24
bit symbol	ENSP13	UVWIS13		AINSP13				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP12	UVWIS12		AINSP12				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP11	UVWIS11		AINSP11				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP10	UVWIS10		AINSP10				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP13	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS13[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP13[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSP12	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS12[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP12[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSP11	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS11[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP11[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSP10	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS10[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP10[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET2:PMD Trigger Program Register 2

	31	30	29	28	27	26	25	24
bit symbol	ENSP23	UVWIS23		AINSP23				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP22	UVWIS22		AINSP22				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP21	UVWIS21		AINSP21				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP20	UVWIS20		AINSP20				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP23	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS23[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP23[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSP22	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS22[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP22[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSP21	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS21[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP21[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSP20	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS20[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP20[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET3:PMD Trigger Program Register 3

	31	30	29	28	27	26	25	24
bit symbol	ENSP33	UVWIS33		AINSP33				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP32	UVWIS32		AINSP32				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP31	UVWIS31		AINSP31				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP30	UVWIS30		AINSP30				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP33	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS33[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP33[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSP32	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS32[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP32[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSP31	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS31[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP31[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSP30	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS30[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP30[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET4:PMD Trigger Program Register 4

	31	30	29	28	27	26	25	24
bit symbol	ENSP43	UVWIS43		AINSP43				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP42	UVWIS42		AINSP42				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP41	UVWIS41		AINSP41				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP40	UVWIS40		AINSP40				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP43	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS43[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP43[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSP42	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS42[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP42[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSP41	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS41[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP41[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSP40	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS40[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP40[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

ADxPSET5:PMD Trigger Program Register 5

	31	30	29	28	27	26	25	24
bit symbol	ENSP53	UVWIS53		AINSP53				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP52	UVWIS52		AINSP52				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP51	UVWIS51		AINSP51				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP50	UVWIS50		AINSP50				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSP53	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	UVWIS53[1:0]	R/W	Phase select (for Vector Engine) See table below.
28-24	AINSP53[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
23	ENSP52	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	UVWIS52[1:0]	R/W	Phase select (for Vector Engine) See table below.
20-16	AINSP52[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
15	ENSP51	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	UVWIS51[1:0]	R/W	Phase select (for Vector Engine) See table below.
12-8	AINSP51[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".
7	ENSP50	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	UVWIS50[1:0]	R/W	Phase select (for Vector Engine) See table below.
4-0	AINSP50[4:0]	R/W	AIN select Refer to "Table 16-2 Select the AIN pin".

Phase select

00	Not specified
01	U
10	V
11	W

16.4.25 ADxTSET03 / ADxTSET47 / ADxTSET811 (Timer Trigger Program Registers)

AD conversion can be started by INTTBx0 generated from TMRBx as a trigger. There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to "1" enables the ADxTSETm register. The <AINSTm[4:0]> are used to select the AIN pin to be used. The numbers of the Timer Trigger Program Registers correspond to those of the AD Conversion Result Registers. When finished this AD conversion, interrupt : INTADxTMR is generated.

(m=0 to 11)

Table 16-3 Select the AIN pin

<AINST0 [4:0]> to <AINST11 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 to 1_1111	:Reserved	

ADxTSET03: Timer Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST3[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
23	ENST2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST2[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
15	ENST1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST1[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
7	ENST0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST0[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".

ADxTSET47: Timer Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST6	-	-	AINST6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST5	-	-	AINST5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST4	-	-	AINST4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST7[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
23	ENST6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST6[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
15	ENST5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST5[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
7	ENST4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST4[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".

ADxTSET811: Timer Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST10	-	-	AINST10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST9	-	-	AINST9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST8	-	-	AINST8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENST11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINST11[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
23	ENST10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINST10[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
15	ENST9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINST9[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".
7	ENST8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINST8[4:0]	R/W	AIN select Refer to "Table 16-3 Select the AIN pin".

16.4.26 ADxSSET03 / ADxSSET47 / ADxSSET811(Software Trigger Program Registers)

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. Setting the <ENSSm> to "1" enables the ADxSSETm register. The <AINSSm[4:0]> are used to select the AIN pin to be used. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers. When finished this AD conversion, interrupt :INTADxSFT is generated.

(m=0 to 11)

Table 16-4 Select the AIN pin

<AINSS0 [4:0]> to <AINSS11 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 to 1_1111	:Reserved	

ADxSSET03: Software Trigger Program Registers 03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS2	-	-	AINSS2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS1	-	-	AINSS1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS0	-	-	AINSS0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS3[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
23	ENSS2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS2[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
15	ENSS1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS1[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
7	ENSS0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS0[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".

ADxSSET47: Software Trigger Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS6	-	-	AINSS6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS5	-	-	AINSS5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS4	-	-	AINSS4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS7[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
23	ENSS6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS6[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
15	ENSS5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS5[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
7	ENSS4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS4[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".

ADxSSET811: Software Trigger Program Registers 811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS10	-	-	AINSS10				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS9	-	-	AINSS9				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS8	-	-	AINSS8				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSS11	R/W	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSS11[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
23	ENSS10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSS10[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
15	ENSS9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSS9[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".
7	ENSS8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSS8[4:0]	R/W	AIN select Refer to "Table 16-4 Select the AIN pin".

16.4.27 ADxASET03 / ADxASET47 / ADxASET811(Constant Conversion Program Registers)

The ADCs allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to "1" enables the ADxASETm register. The <AINSAm[4:0]> are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m=0 to 11)

Table 16-5 Select the AIN pin

<AINSA0 [4:0]> to <AINSA11 [4:0]>	ADC Unit A	ADC Unit B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 to 1_1111	:Reserved	

ADxASET03: Constant Conversion Program Registers03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA2	-	-	AINSA2				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA1	-	-	AINSA1				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA0	-	-	AINSA0				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA3	R/W	ADxREG3 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA3[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
23	ENSA2	R/W	ADxREG2 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA2[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
15	ENSA1	R/W	ADxREG1 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA1[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
7	ENSA0	R/W	ADxREG0 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA0[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".

ADxASET47: Constant Conversion Program Registers 47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA6	-	-	AINSA6				
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA5	-	-	AINSA5				
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA4	-	-	AINSA4				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31	ENSA7	R/W	ADxREG7 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA7[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
23	ENSA6	R/W	ADxREG6 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA6[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
15	ENSA5	R/W	ADxREG5 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA5[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
7	ENSA4	R/W	ADxREG4 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA4[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".

ADxASET811: 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	ADxREG11 enable 0:Disable 1:Enable
30-29	-	R	Read as "0".
28-24	AINSA11[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
23	ENSA10	R/W	ADxREG10 enable 0:Disable 1:Enable
22-21	-	R	Read as "0".
20-16	AINSA10[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
15	ENSA9	R/W	ADxREG9 enable 0:Disable 1:Enable
14-13	-	R	Read as "0".
12-8	AINSA9[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".
7	ENSA8	R/W	ADxREG8 enable 0:Disable 1:Enable
6-5	-	R	Read as "0".
4-0	AINSA8[4:0]	R/W	AIN select Refer to "Table 16-5 Select the AIN pin".

16.5 Operation Descriptions

16.5.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the VREFHA and VREFLA pins are used in ADC A and the VREFHB and VREFLB pins are used in ADC B. When ADxMOD3<RCUT> is set to "1", VREFHx-VREFLx is switched off from the condition switched on.

16.5.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD trigger (See "16.4.24 PMD Trigger Program Registers")
- Timer trigger (TMRB) (See "16.4.25 Timer Trigger Program Registers.")
- Software trigger (See "16.4.26 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.

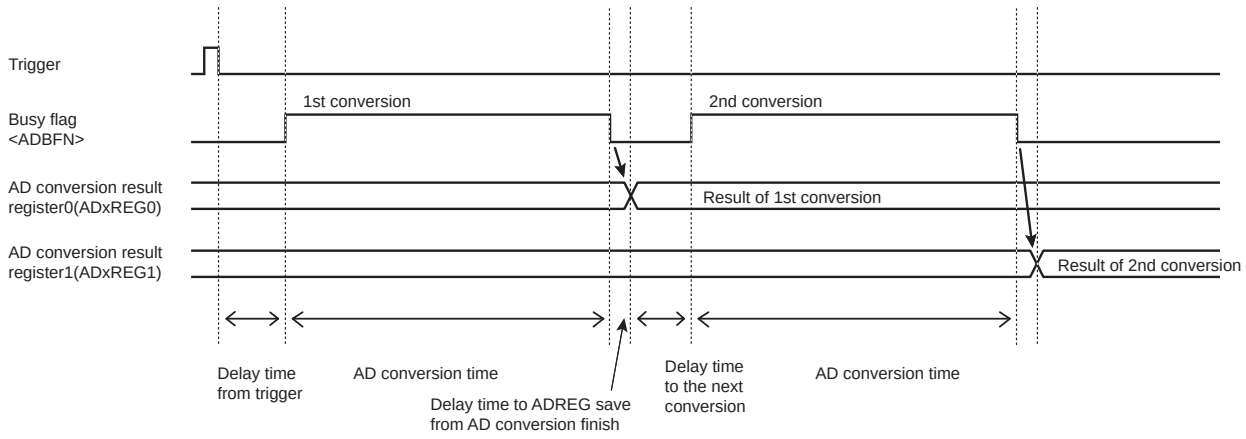


Figure 16-3 Timing chart of AD conversion

Table 16-6 AD conversion time and delay time

Trigger	Delay time from trigger (Note 1)		conversion time	Delay time to ADREG save from AD conversion finish	Delay time to the next conversion (Note 2)	
	Min	Max			Min	Max
PMD [1/SCLK]	5	8	120	2	3	5
TMRB [1/SCLK]		16				13
Software [1/SCLK]	6	17				
Constant [1/SCLK]						

Note 1: Delay time from trigger to conversion start

Note 2: Delay time from ADREG save to 2nd conversion start and more than one conversion start, with identical trigger

16.5.3 AD Conversion Monitoring Function

The ADCs have the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function, set ADxCMPCR0<CMP0EN> or ADxCMPCR1<CMP1EN> to "1". In the monitoring function, if the value of AD conversion result register to which the monitoring function is assigned corresponds to the comparison condition specified by ADxCMCR0<ADBIG0>/ADxCMPR1<ADBIG1>, the interrupt (INTADxCPA for the monitoring function 0, INTADxCPB for the monitoring function 1) 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.

16.6 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

16.6.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADxSSET03, ADxSSET47 and ADxSSET811.(Figure 16-4)

If the ADxMOD1<ADEN> is cleared to "0" during AD conversion, the ongoing conversion stops without storing to the result register.(Figure 16-5)

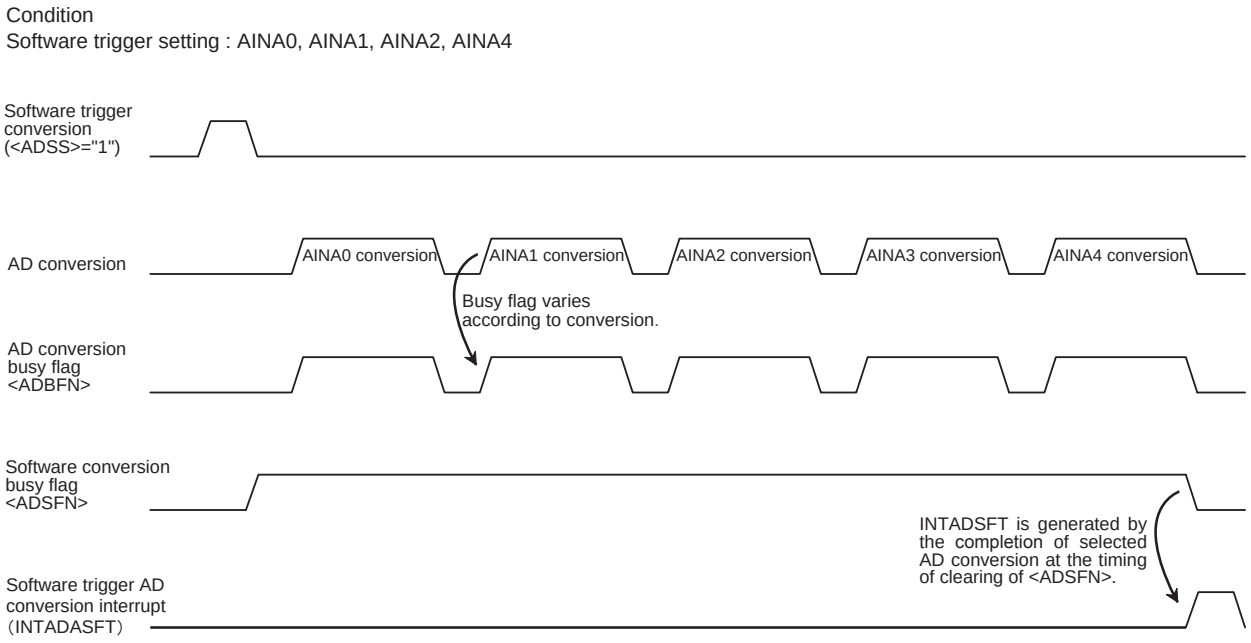


Figure 16-4 Software trigger AD conversion

Condition
Software trigger setting : AINA0, AINA1, AINA2

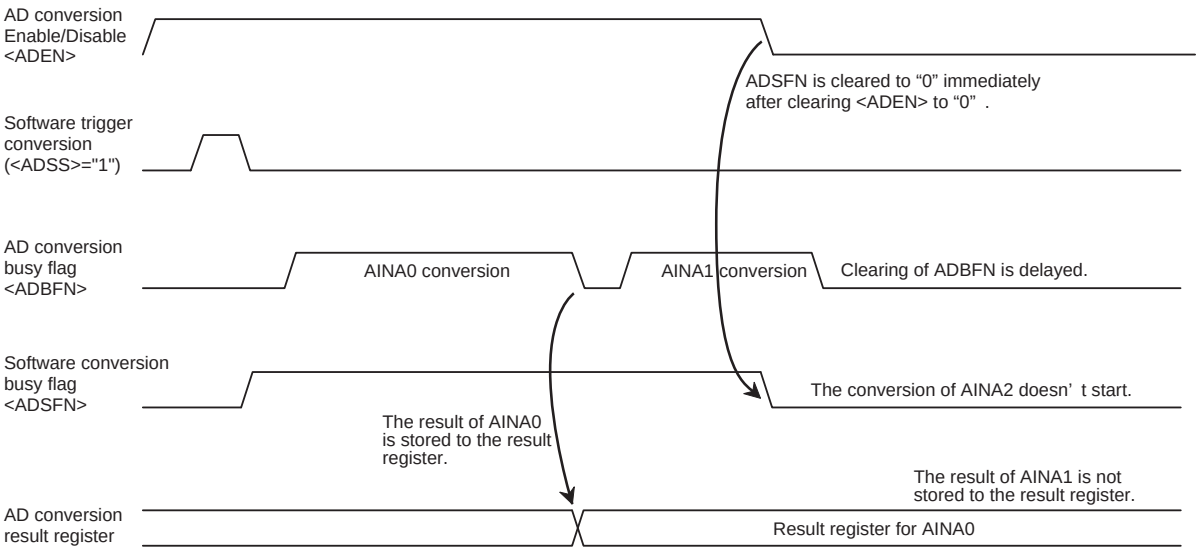


Figure 16-5 Writing "0" to <ADEN> during the software trigger AD conversion

16.6.2 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 16-6)

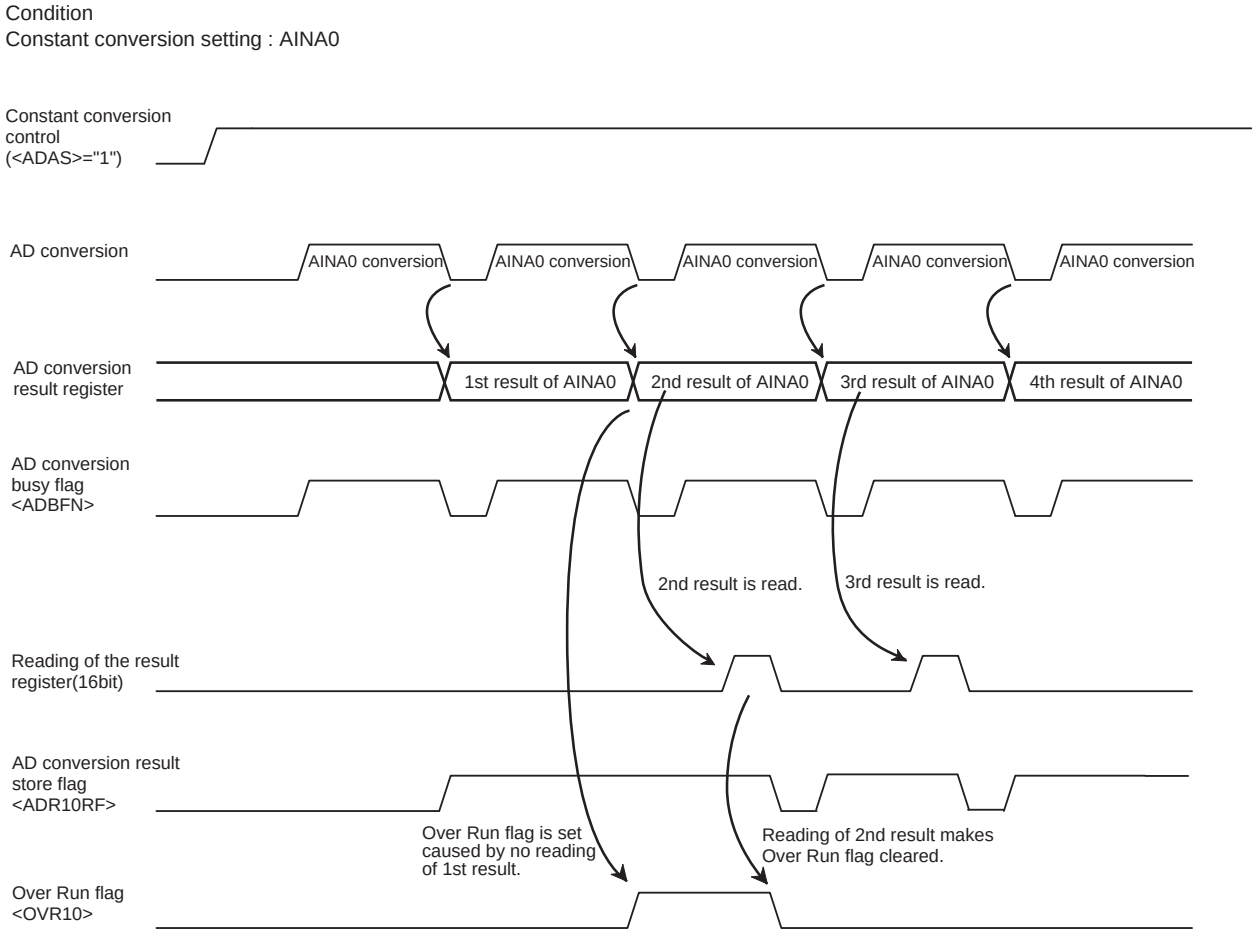


Figure 16-6 Constant conversion

16.6.3 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 16-7) 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 16-8)

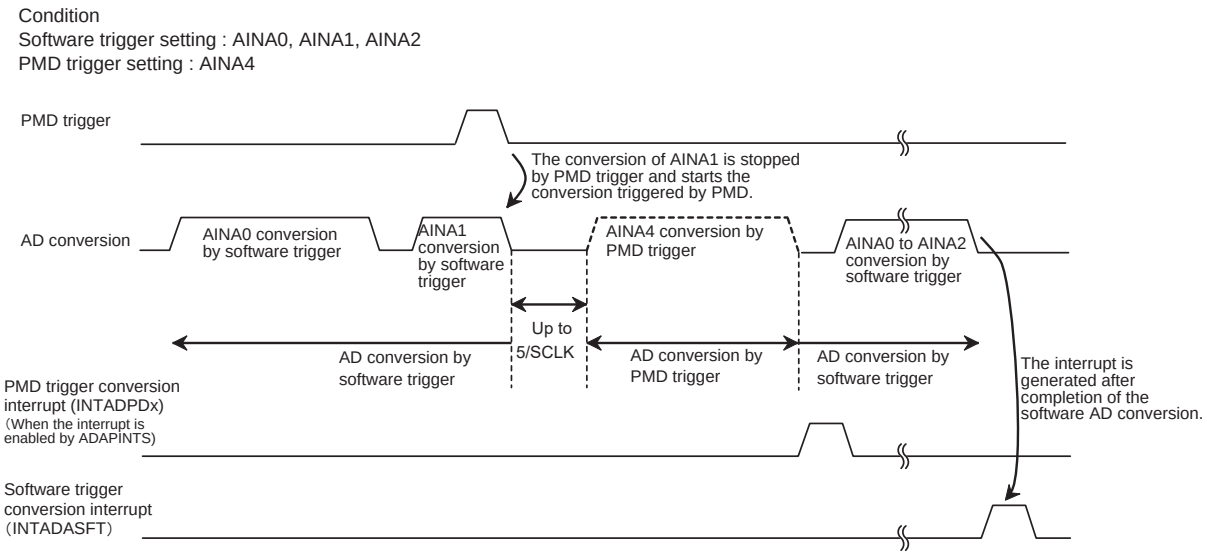


Figure 16-7 AD conversion by PMD trigger

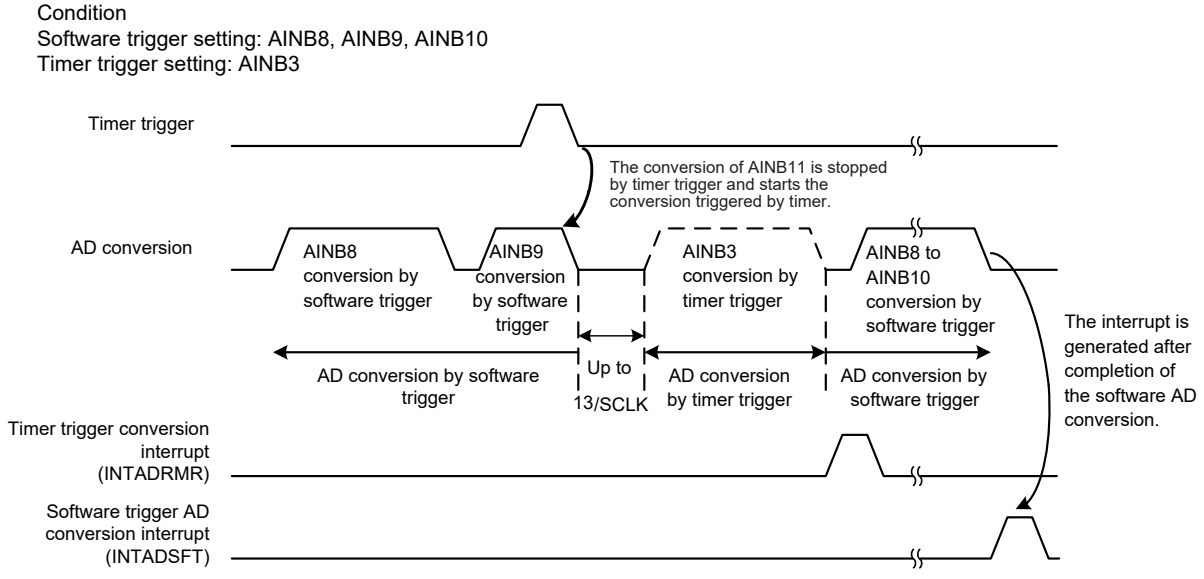


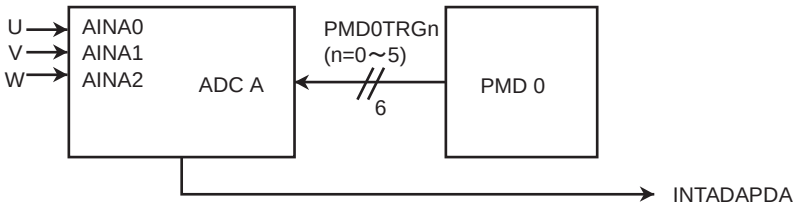
Figure 16-8 AD conversion by timer trigger

Note: When timer trigger is not used, do not use INTTB51. Set TB5IM<TBIM1> to "1".

16.7 Usage Examples

16.7.1 Successive Conversion Using One PMD(Three Shunts) and One ADC

The following shows a circuit diagram for AD conversion using one PMD0 for three shunts and one ADC.



Example ADC settings are shown below.

ADC UnitA

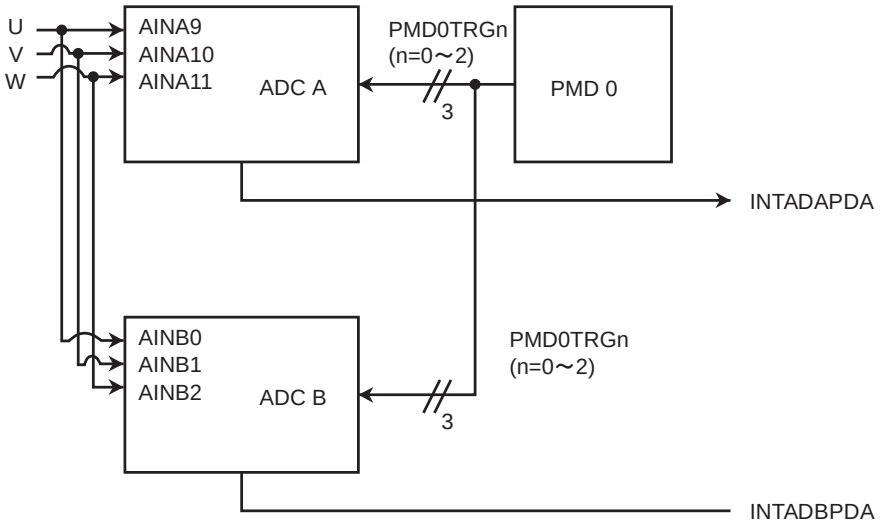
Program	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	A	A	A	A	A	A

Programs 0 to 5 are assigned to trigger inputs PMD0TRG0 to 5. "reg0" and "reg1" indicate the PMD Trigger Program Registers ADAPSETn[7:0] and ADAPSETn[15:8]. "U", "V" and "W" indicate the phases of a motor. AIN inputs are selected to obtain these phases.

When a trigger input occurs, AD conversion is performed based on reg0 and reg1 sequentially, and then the interrupt signal (INTADAPDA) is generated.

16.7.2 Simultaneous Conversion Using One PMD (Three Shunts) and Two ADCs

The following shows a block diagram for AD conversion using PMD0 for three shunts and two ADCs.



Example ADC settings are shown below.

ADC UnitA

Program	0	1	2
reg0	U	V	W
INT	A	A	A

ADC UnitB

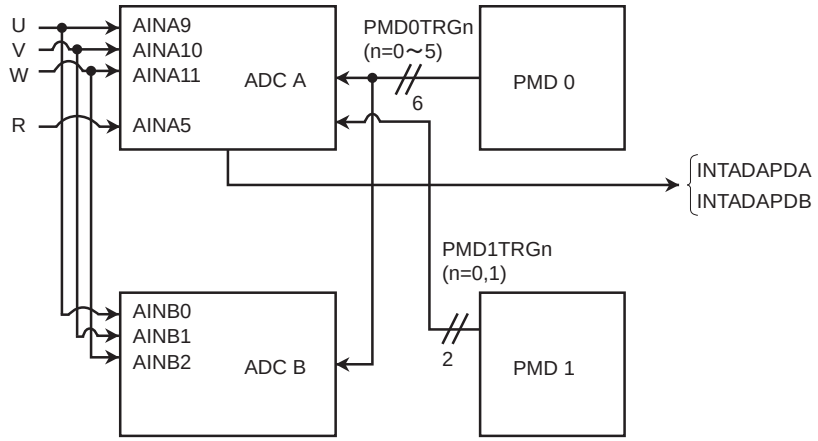
Program	0	1	2
reg0	V	W	U
INT	A	A	A

Programs 0 to 2 are assigned to three trigger inputs to ADC A and ADC B. "reg0" indicates the PMD Trigger Program Register ADAPSETn[7:0] and ADBPSETn[7:0]. "U", "V" and "W" indicate the phases of a motor. AIN inputs are selected to obtain these phases.

When a trigger input occurs, ADC A and ADC B are started simultaneously to perform AD conversion based on reg0, and the interrupt signals (INTADAPDA , INTADBPDA) are output to ADC A and ADC B.

16.7.3 Simultaneous Conversion Using PMD0 (Three Shunts), PMD1 (One Shunt) and Two ADCs

The following shows a circuit diagram for AD conversion using PMD0 for three shunts, PMD1 for one shunt and two ADCs.



Example ADC settings are shown below.

ADC UnitA

Trigger	PMD0	PMD0	PMD0	PMD1	PMD1
	0,3	1,4	2,5	6	7
Program	0	1	2	3	4
reg0	U	V	W	-	-
reg1	-	-	-	R	-
reg2	-	-	-	-	R
INT	A	A	A	-	B

ADC UnitB

Trigger	PMD0	PMD0	PMD0
	0,3	1,4	2,5
Program	0	1	2
reg0	V	W	U
INT	-	-	-

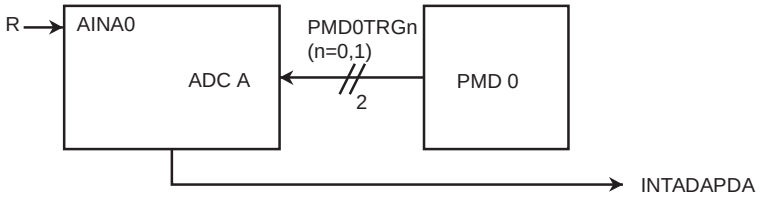
In ADC A, programs 0 to 2 are assigned to six trigger signals from PMD0 and programs 3 and 4 are assigned to two trigger signals from PMD1. In ADC B, programs 0 to 2 are assigned to six trigger signals from PMD0.

"reg0", "reg1" and "reg2" indicate the PMD Trigger Program Registers ADxPSETn[7:0], ADxPSETn [15:8] and ADxPSETn[23:16] (x=A,B : ADC Unit). "U", "V" and "W" indicate the phases of a motor. AIN inputs are selected to obtain these phases. "R" indicates a resistor, where the AIN that is connected to that resistor is set.

When a trigger input occurs, ADC A or ADC B is started to perform AD conversion. In ADC A, the interrupt (INTADAPDA) is generated for a trigger from PMD0 and the interrupt (INTADAPDB) is generated for a trigger from PMD1. In ADC B, interrupt generation is disabled in this example.

16.7.4 Successive Conversion Using One PMD (One Shunt) and One ADC

The following shows a circuit diagram for AD conversion using PMD0 for one shunt and one ADC.



Example ADC settings are shown below.

ADC UnitA

Trigger	PMD0	PMD0
	0	1
Program	0	1
reg0	R	-
reg1	-	R
INT	-	A

Programs 0 and 1 are assigned to two trigger signals from PMD0.

"reg0" and "reg1" indicate the PMD Trigger Program Registers ADAPSETn[7:0] and ADAPSETn[15:8]. "R" indicates a resistor, where the AIN input that is connected to that resistor is set.

When a trigger input occurs, the ADC is started to execute programs 0 and 1 sequentially. When program 1 is completed, the interrupt (INTADAPDA) is generated.

16.8 Precautions on Use of AD Converter

During AD conversion, do not change the output data of port H/J/P, 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.

17. 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 (A-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.

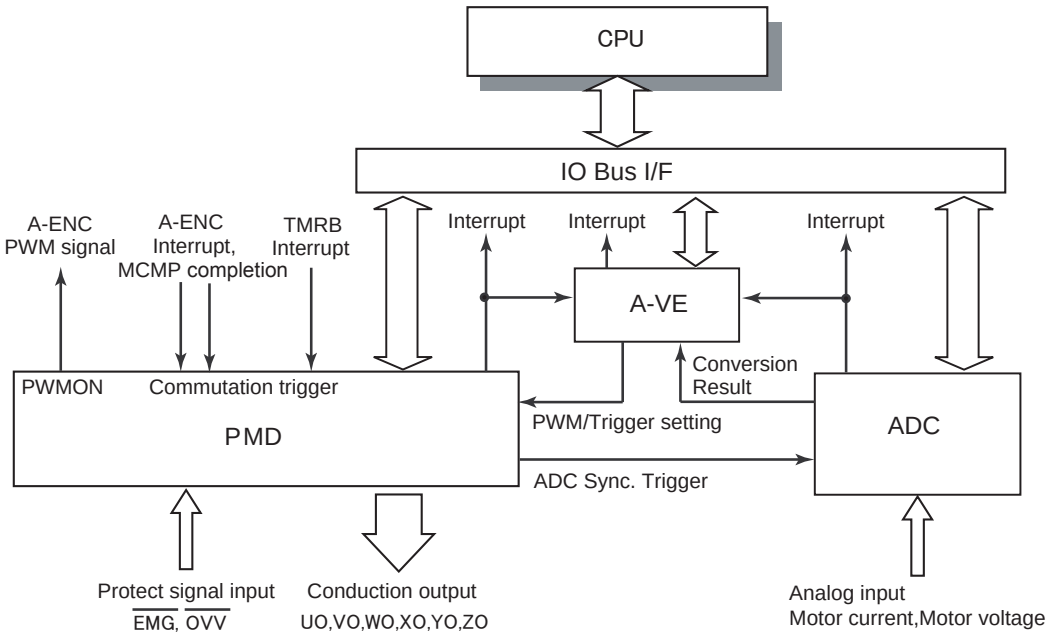


Figure 17-1 Block Diagram of Functions related to Motor Control

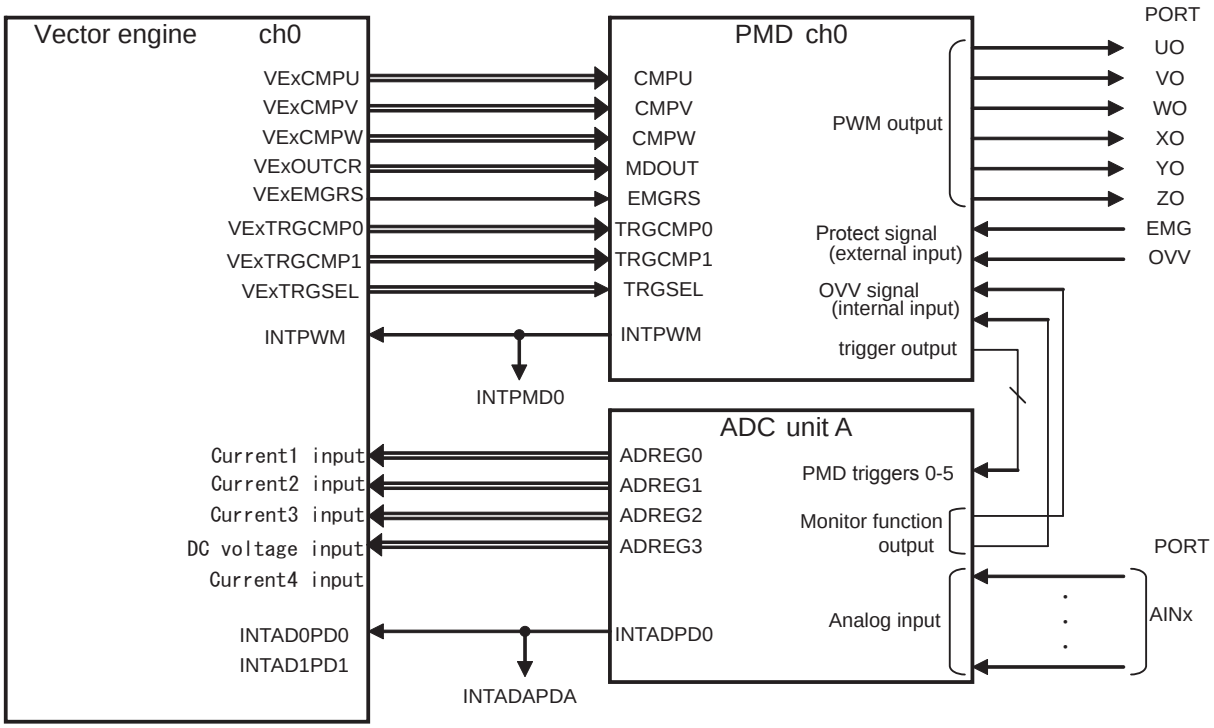


Figure 17-2 Related diagram of Motor control circuit, Vector engine and A/D converter

17.1 PMD Circuit configuration

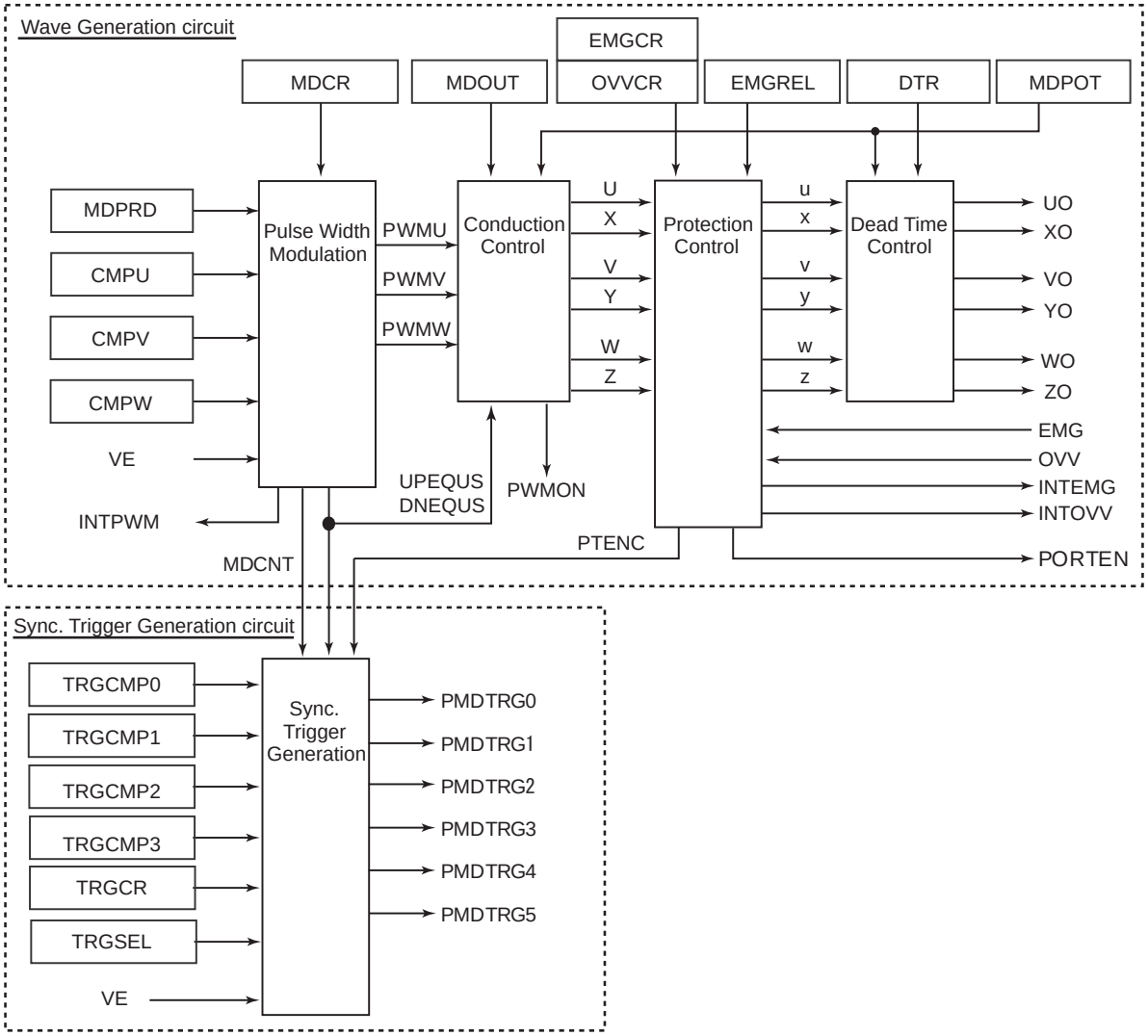


Figure 17-3 Block diagram of PMD Circuit

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.

17.2 PMD Registers

The following table lists the control registers and their addresses:

For the base address, refer to "A list of peripheral function base addresses" 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

17.2.1 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.

17.2.2 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]>.

17.2.3 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: VExTRGSEL)
2	MDSEL2	R/W	Mode Select 2 0: Bus mode (using PMD registers: PMDxTRGCMP0 and PMDxTRGCMP1) 1: VE mode (using VE registers: VExTRGCMP0 and VExTRGCMP1)
1	MDSEL1	R/W	Mode Select 1 0: Bus mode (using PMD register: PMDxMDOUT) 1: VE mode (using VE register: VExOUTCR)
0	MDSEL0	R/W	Mode Select 0 0: Bus mode (using PMD registers: PMDxCMPU,PMDxCMPV1 and PMDxCMPW) 1: VE mode (using VE registers: VExCMPU,VExCMPV and VExCMPW registers and sets VExEMGRS register enabled)

17.2.4 Pulse Width Modulation Circuit

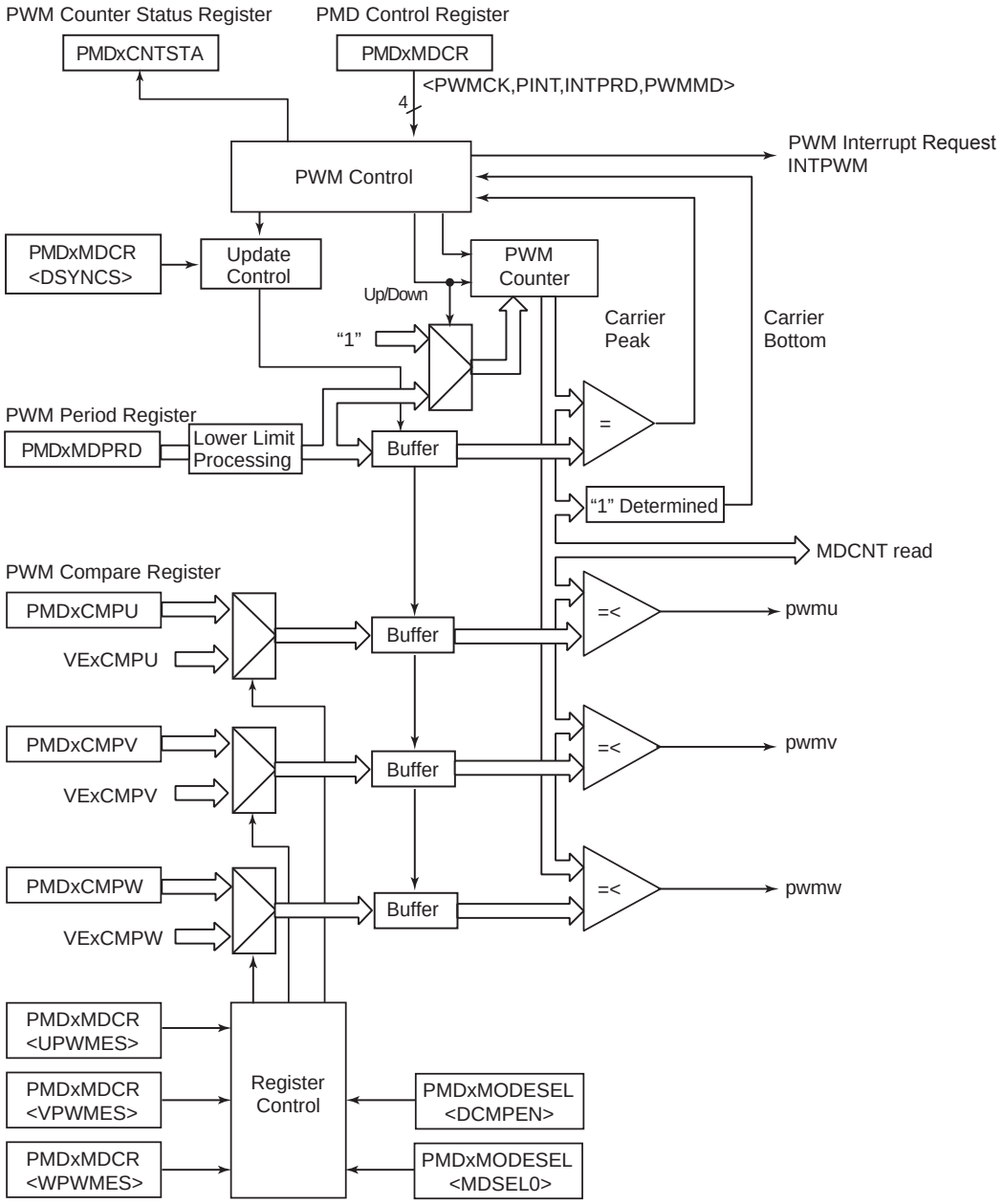


Figure 17-4 Pulse Width Modulation Circuit

The pulse width modulation circuit has a 16-bit PWM up-/down-counter and generates PWM carrier waveforms with a resolution of 1/fsys (8.33[ns] at 120[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 (33.3[ns] at 120[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 17-5 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 17-6 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W Buffer Update Timing".)

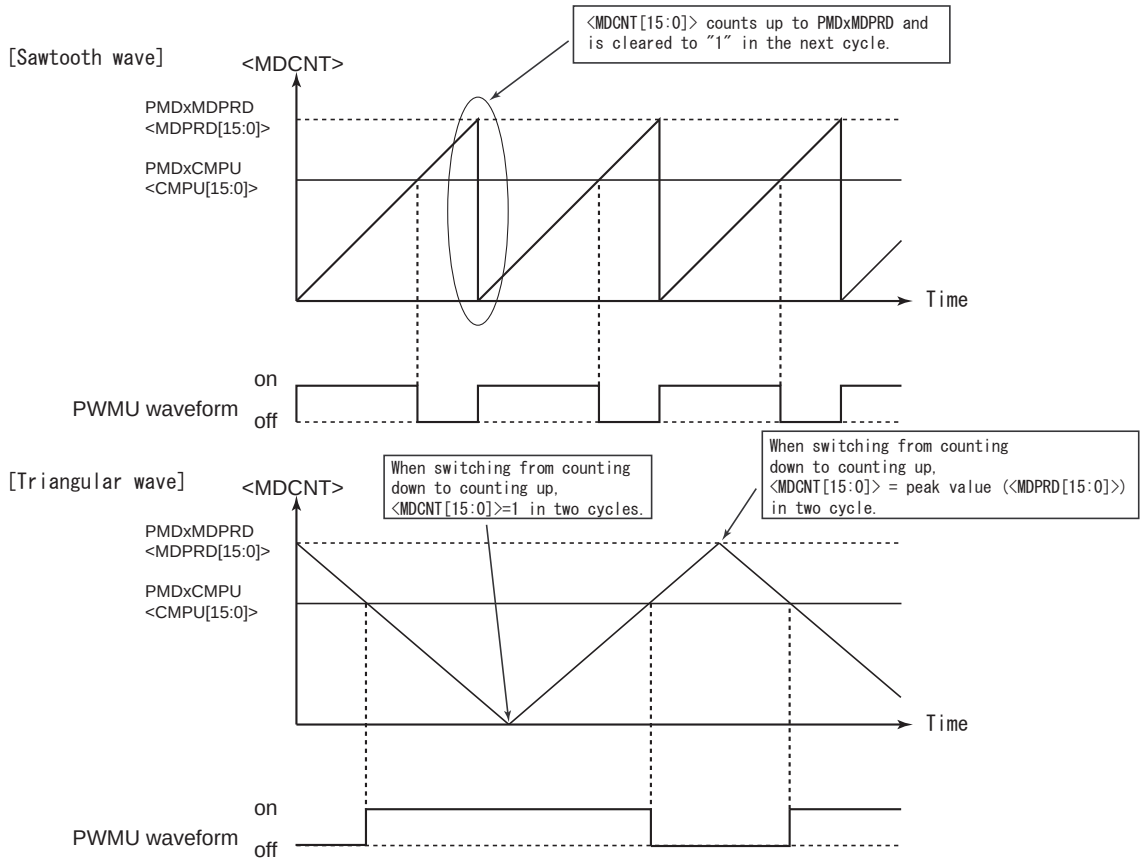


Figure 17-5 PWM Waveforms

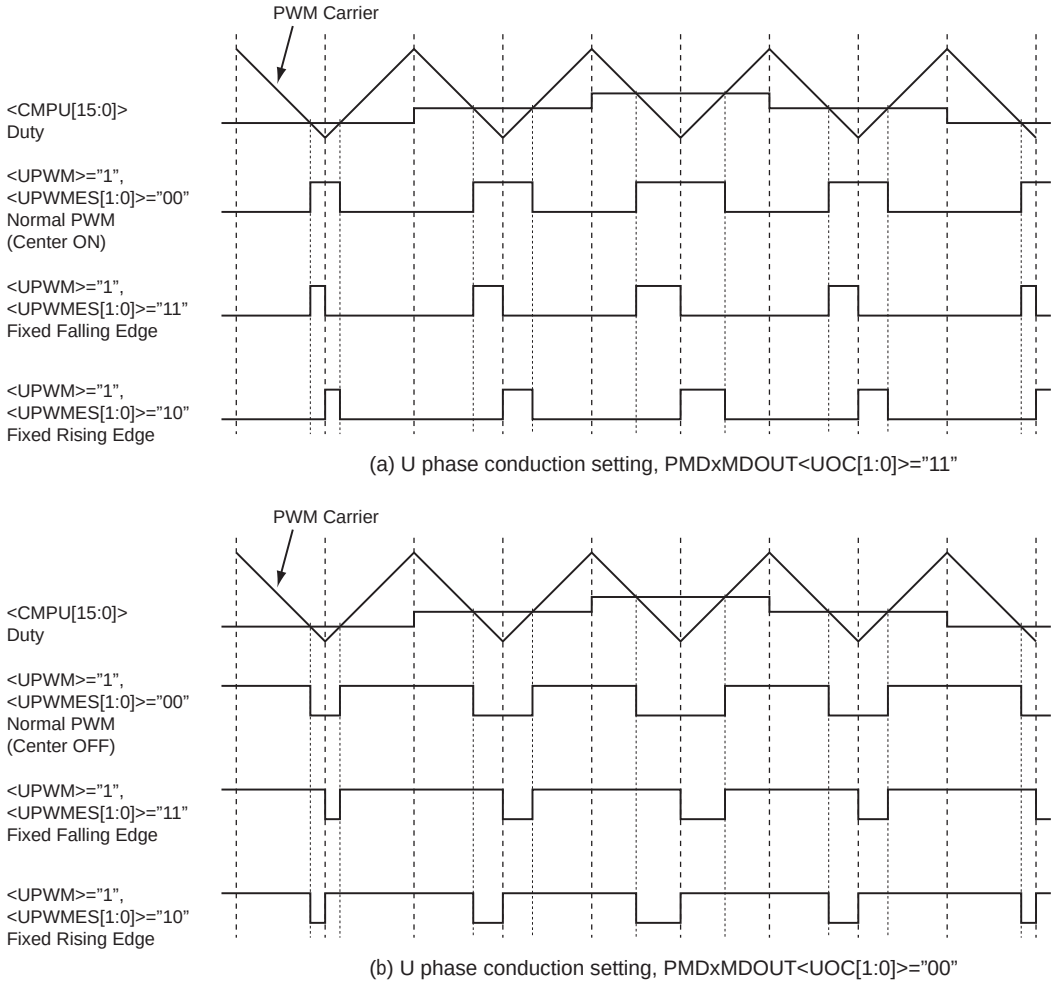


Figure 17-6 Waveforms of PWM triangular wave carrier using fixed edge

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.

17.2.4.1 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 17-1) 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 (8.33[ns] at 120[MHz]) / triangular wave 2/fsys (16.7[ns] at 120[MHz]) Quadruple cycle setting: sawtooth wave 4/fsys (33.3[ns] at 120[MHz]) / triangular wave 8/fsys (66.7[ns] at 120[MHz])
5	SYNTMD	R/W	Port output mode Port outputs are controlled by a combination of <nOC>,<nPWM>,<POLH>,<POLL> and <SYNTMD> (refer to Table 17-4).

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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W 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 17-2 Switching control of PMDxCMPU/V/W and VExCMPU/V/W

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

17.2.4.2 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.

17.2.4.3 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 (8.33[ns] at 120[MHz]) Note1: When a quadruple cycle mode is selected (PMDxMDCR<PWMCK>="1"), time resolution of the counter is 4/fsys (33.3[ns] at 120[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]>

17.2.4.4 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W Buffer Update Timing".

Note 4: Read value is the first buffer value (the latest data set via a bus).

17.2.4.5 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W Buffer Update Timing".

Note 5: Read value is the first buffer value (the latest data set via a bus).

17.2.4.6 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W Buffer Update Timing".

Note 5: Read value is the first buffer value (the latest data set via a bus).

17.2.4.7 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W Buffer Update Timing".

Note 5: Read value is the first buffer value (the latest data set via a bus).

17.2.5 Conduction Control Circuit

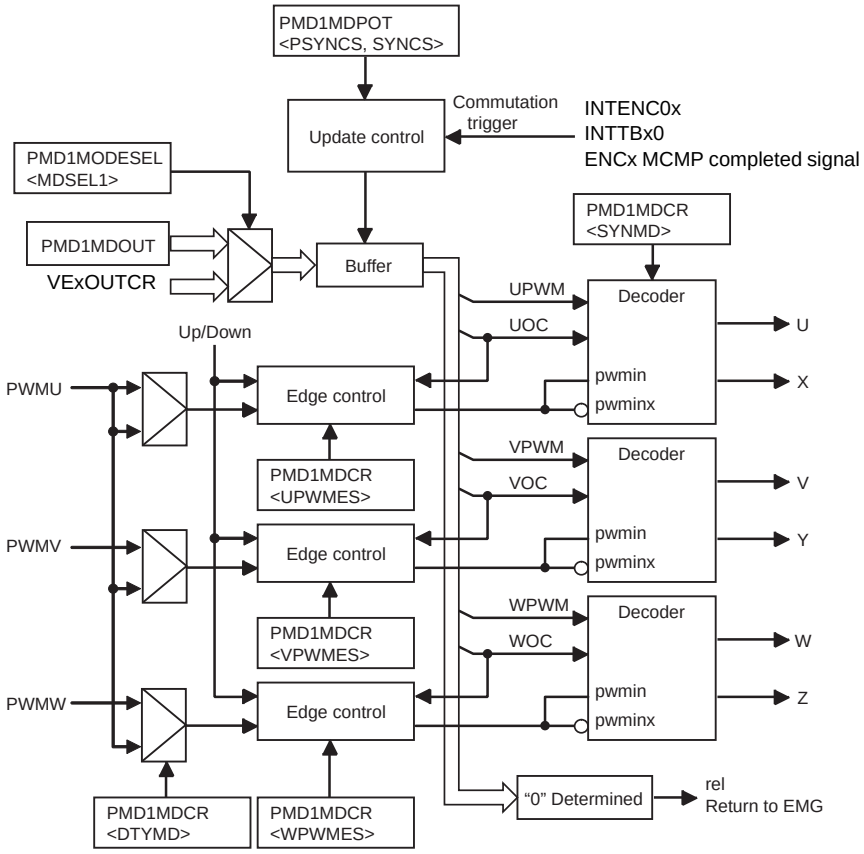


Figure 17-7 Conduction Control Circuit

The conduction control circuit performs the output port control according to the settings made in the output control register PMDxMDOUT(VExOUTCR) and the output setting register PMDxMDPOT. PMDxMDOUT (VExOUTCR) 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 17-3 Update Timing of the PMDxMDOUT(VExOUTCR) buffer".)

Using PMDxMDPOT<POLH>,<POLL>, six output ports can be set to low-active or high-active on upper-phase output (UO,VO,WO) or lower-phase output (XO,YO,ZO) respectively. In addition, <WPWM>, <VPWM>, <UPWM> of the PMDxMDOUT(VExOUTCR) 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(VExOUTCR).

"Table 17-4 Port Outputs according to the <UOC>,<VOC>,<WOC>,<UPWM>,<VPWM> and <WPWM> settings" shows port outputs setting according to port output setting in the PMDxMDOUT(VExOUTCR) register and PMDxMDPOT register, and port output polarity setting in the port output mode of PMDxMDCR register.

The conduction control circuit outputs a PWM signal (PWMON) to the encoder input circuit (A-ENC). The external input signal in the A-ENC is sampled synchronously with the PWMON.

Table 17-3 Update Timing of the PMDxMDOUT(VExOUTCR) 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	When INTENC0x is arisen.	The first PWM carrier bottom every time when INTENC0x is arisen.	The first PWM carrier peak every time when INTENC0x is arisen.	Either the first PWM carrier peak or the first carrier bottom every time when INTENC0x is arisen.
	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	When MCMP completed signal from ENCx (CTRGO) is arisen.	The first PWM carrier bottom every time when CTRGO is arisen.	The first PWM carrier peak every time when CTRGO is arisen.	Either the first PWM carrier peak or the first carrier bottom every time when CTRGO is arisen.

Note: IF PMD is disabled (PMDxMDCR<PMWEN>="0"), the retained trigger condition is cleared.

17.2.5.1 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	-	-	-	-	POLH	POLL	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(VExOUTCR) transfer timing (trigger synchronous setting). 00: asynchronous 01: when INTENCx (ENCx interrupt request) occurs 10: when INTTBx0 (TMRBx interrupt request) occurs 11: when CTRGO(ENCx MCMP completed) occurs 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 17-3 Update Timing of the PMDxMDOUT(VExOUTCR) buffer"). Note2: When PMD is disabled (PMDxMDEN<PWMEN>="0"), the timing is asynchronous regardless of settings.
7-4	−	R	Read as 0.
3	POLH	R/W	Selects the output polarity of the upper phase output (UO, VO, WO). 0: Active low 1: Active high
2	POLL	R/W	Selects the output polarity of the lower phase output (XO, YO, ZO). 0: Active low 1: Active high
1-0	PSYNCS[1:0]	R/W	Selects PMDxMDOUT(VExOUTCR) transfer timing (PWM synchronous setting). 00: asynchronous to PWM The setting is applied to the port output at the same time that the PMDxMDOUT/VExOUTCR 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 17-3 Update Timing of the PMDxMDOUT(VExOUTCR) 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".

17.2.5.2 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>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 17-4).
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>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 17-4).
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>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 17-4).
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>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 17-4).
3-2	VOC[1:0]	R/W	V-phase conduction control setting Port output is controlled by the combination of <VOC>,<VPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 17-4).
1-0	UOC[1:0]	R/W	U-phase conduction control setting Port output is controlled by the combination of <UOC>,<UPWM>,<POLH>,<POLL> and <SYNTMD> (refer to the Table 17-4).

Note 1: To load the subsequent stage buffer with the value in the PMDxMDOUT(VExOUTCR) 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 17-3 Update Timing of the PMDxMDOUT (VExOUTCR) buffer".

Note 5: Read value is the first buffer value (the latest data set via a bus).

Table 17-4 Port Outputs according to the <UOC>,<VOC>,<WOC>,<UPWM>,<VPWM> and <WPWM> settings

PMDxMDCR<SYNTMD>="0"

Polarity: Active high (PMDxMDPOT<POLH><POLL>="11")

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	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMDxMDCR<SYNTMD>=0

Polarity: Active low (PMDxMDPOT<POLH><POLL>="00")

PMDxMDOUT 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	H	PWM	$\overline{PWM}$
0	1	H	L	H	$\overline{PWM}$
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

PMDxMDCR<SYNTMD>=1

Polarity: Active high (PMDxMDPOT<POLH><POLL>="11")

PMDxMDOUT 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	L	L	$\overline{PWM}$	PWM
0	1	L	H	L	$\overline{PWM}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{PWM}$

PMDxMDCR<SYNTMD>=1

Polarity: Active low (PMDxMDPOT<POLH><POLL>="00")

PMDxMDOUT 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	H	PWM	$\overline{PWM}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{PWM}$	H
1	1	L	L	$\overline{PWM}$	PWM

17.2.6 Protection Control Circuit

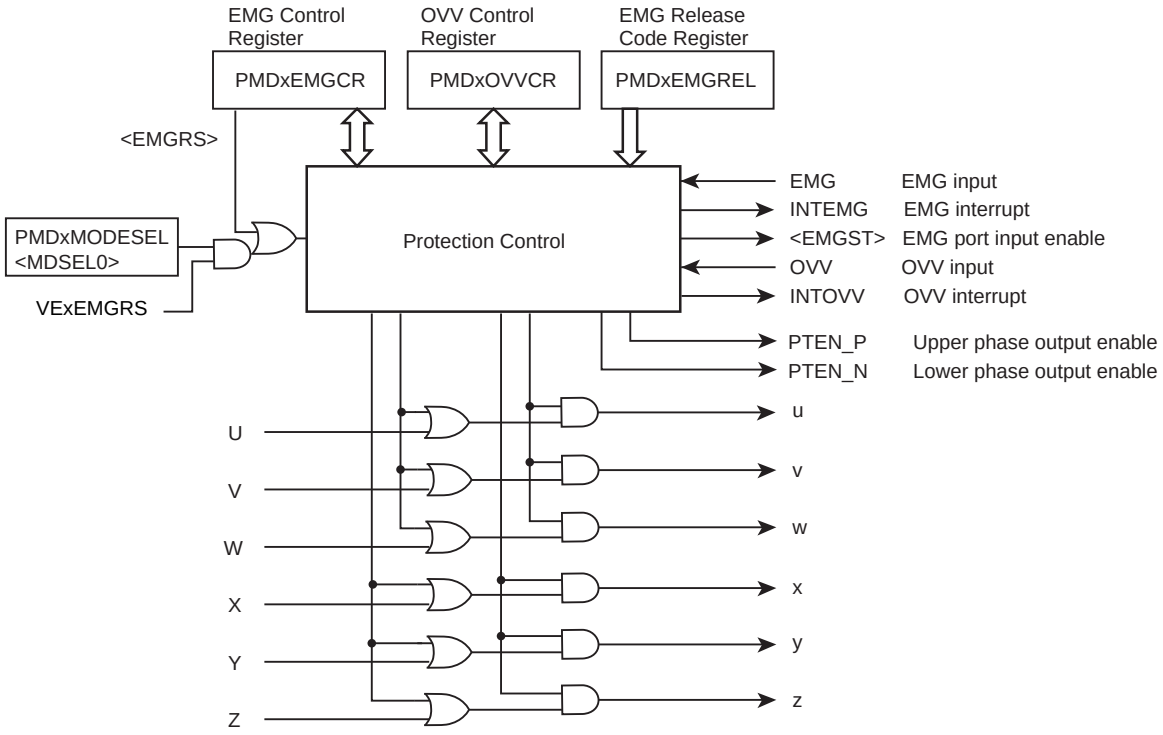


Figure 17-8 Protection Control Circuit

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

17.2.6.1 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(VExOUTCR)<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC>.) (Note1) and then setting either PMDxEMGCR<EMGRS> or VExEMGRS<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(VExOUTCR) 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(VExOUTCR)<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC> to "0" to make all ports in-active.
4: Releases EMG protection by setting PMDxEMGCR(VExEMGRS)<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.

17.2.6.2 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.

17.2.6.3 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: 133[ns] at 120[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(VExOUTCR) 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(VExOUTCR) 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.

17.2.6.4 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.

17.2.6.5 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.

17.2.6.6 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: 133[ns] at 120[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. Active and inactive are depends on the settings of <POLL> and <POLH>. 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 17-1 PMDxMDPRD, PMDxCMPU/V/W and VExCMPU/V/W 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.)

17.2.6.7 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.

17.2.7 Dead Time Control Circuit

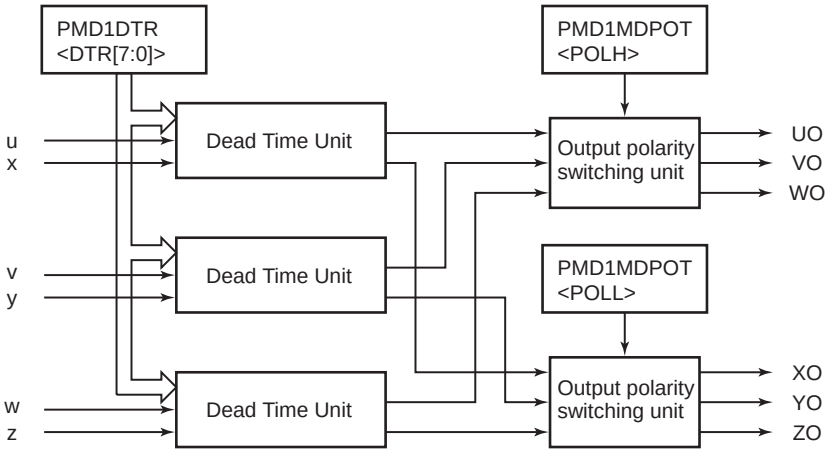


Figure 17-9 Dead Time Control Circuit

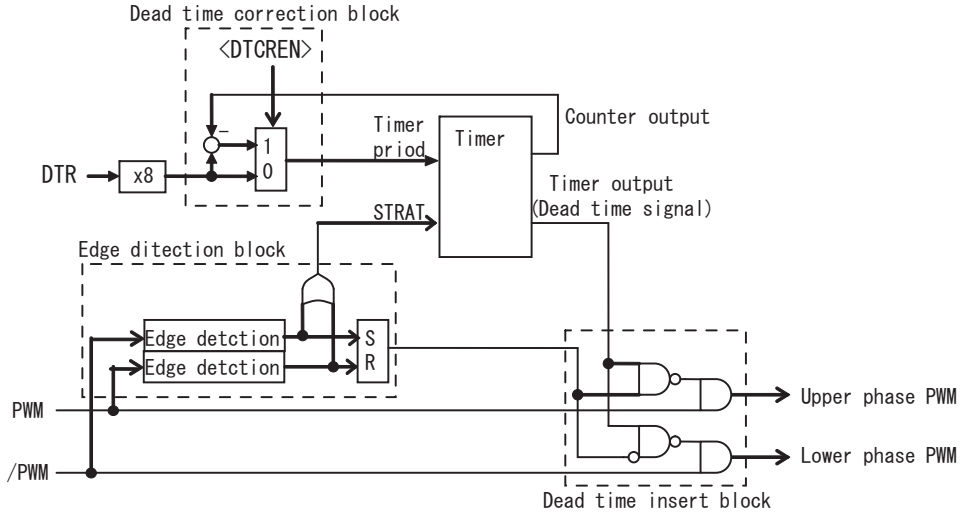


Figure 17-10 The dead time unit configuration

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 17-10 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 (66.7[ns] at 120[MHz]) can be set.

The output polarity switching circuit allows the polarity (active high or active low) of the upper-output (UO, VO, WO) and lower-output (XO,YO,ZO) phases to be independently set through PMD output setting register PMDxMDPOT<POLH> and <POLL>.

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 17-11 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%.

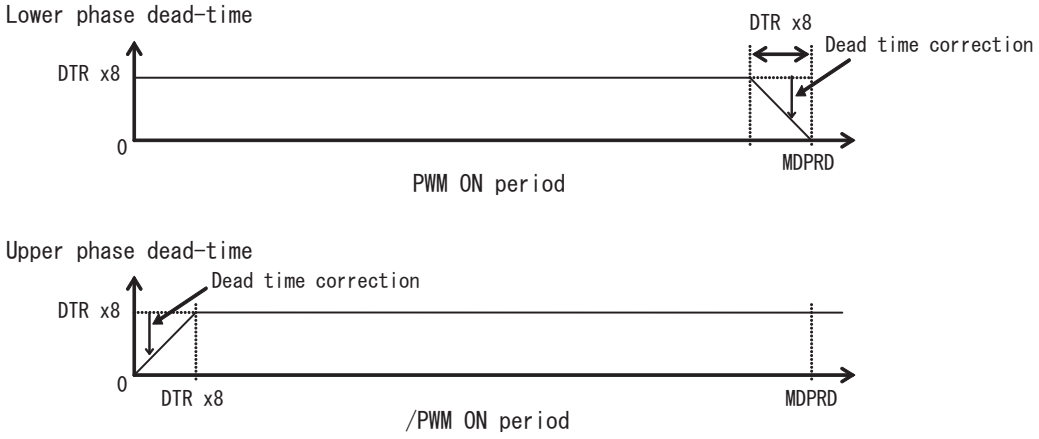


Figure 17-11 Dead Time correction

17.2.7.1 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 17[μs] at 120[MHz])

17.2.8 Sync Trigger Generation Circuit

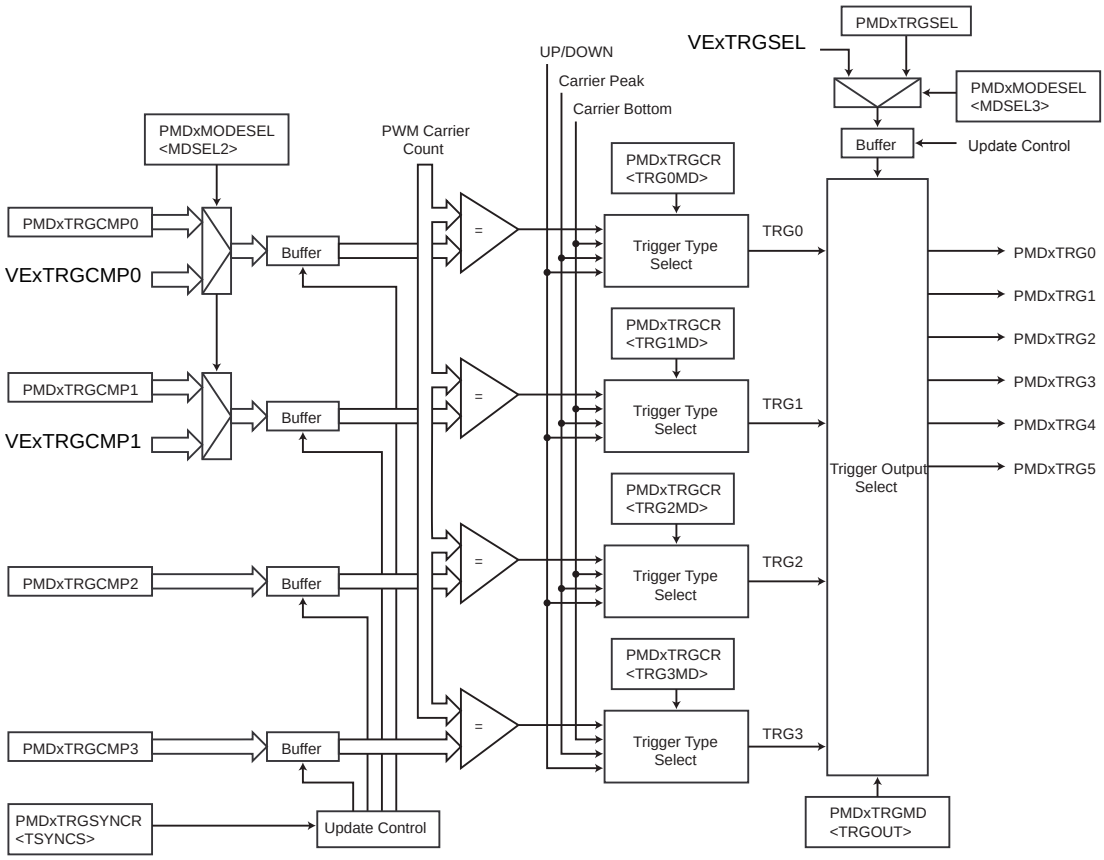


Figure 17-12 Sync Trigger Generation Circuit

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 VExTRGCMP0, VExTRGCMP1 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 (VExTRGSEL). The TRG0 setting is set by PMDxTRGCMP0 (VExTRGCMP0) 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.

17.2.8.1 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 17-5 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 17-5 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") .

17.2.8.2 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 17-5 Buffer update timing for the trigger compare register".

Note 5: Read value is the first buffer value (the latest data set via a bus).

17.2.8.3 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 17-5 Buffer update timing for the trigger compare register".

Note 4: Read value is the first buffer value (the latest data set via a bus).

17.2.8.4 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 17-5 Buffer update timing for the trigger compare register"

Note 4: Read value is the first buffer value (the latest data set via a bus).

17.2.8.5 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 17-5 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 17-5 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 17-5 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 17-5 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.

17.2.8.6 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 17-5 Buffer update timing for the trigger compare register". Note: Asynchronous update PMDxMDEN<PWMEN>="0" regardless of setting.

17.2.8.7 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 17-6 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 17-6 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

17.2.8.8 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 17-6.)

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.

18. Vector Engine (A-VE)

18.1 Outline

18.1.1 Features

1. Basic tasks for vector control (coordinate transformation, phase transformation and SIN/COS computation) are executed on fixed-point format data.

   →It is not necessary for software process to manage the decimal point alignment.

2. Interface processes (output control, trigger generation and input process) for the motor control circuit (PMD) and the AD converter (ADC) are incorporated.
   - 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. Values of current, voltage and rotation speed are normalized with in their maximum values and are computed.

   A decimal point number is used on the fixed-point format.

4. PI control is incorporated for current control.

5. Phase interpolation is implemented to integrate the rotational speed.

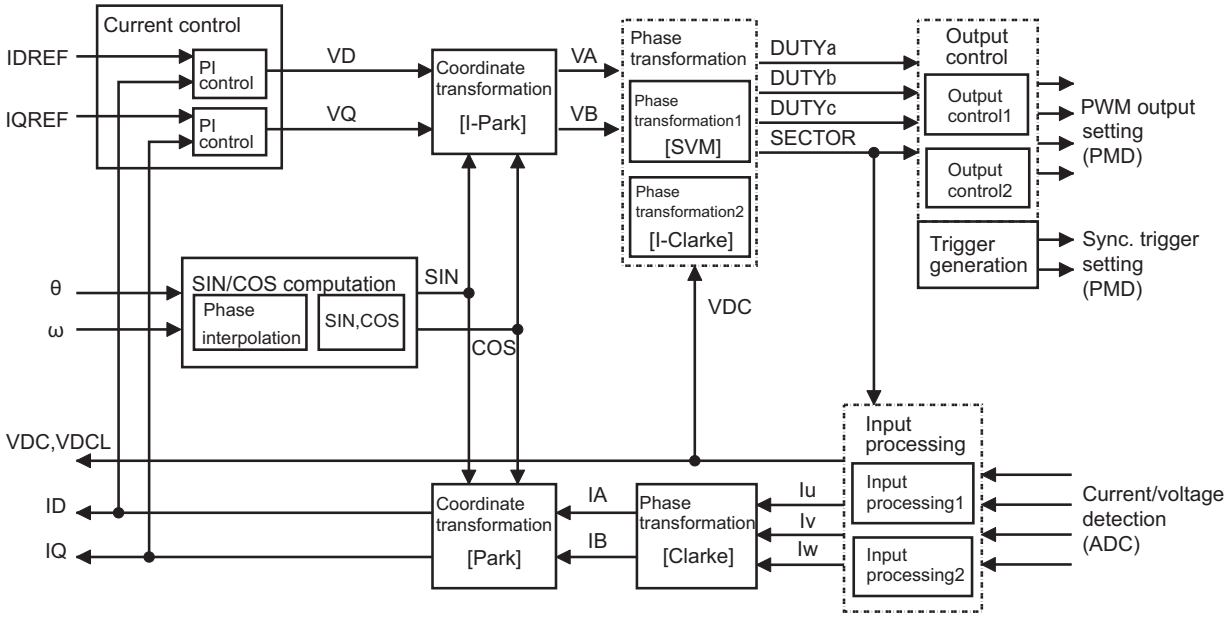


Figure 18-1 Block diagram of the vector control circuit

18.1.2 Major Functions

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. A trigger generation function can generate ADC sampling timing for sensorless current detection. This sampling timing can be used on 1-shunt current detection method.

3. In current control, PI control corresponds d-axis and q-axis independently. It is also possible to directly set d-axis and q-axis voltage registers.

   Controls the result of PI control

   Controls d-axis and q-axis independently.

   Controls an output using the value of voltage scalar consisting of d-axis and q-axis.

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.

   As phase information, direct setting or phase interpolation that integrates the rotational speed on the PWM cycle can be used. When phase information is performed, clipping can be used at the specified phase.

   In output control, dead time compensation and PWM output control are available.

Note: It is necessary to set the motor control circuits and the ADC when the vector engine is used.

- When the vector engine is used, set the VE mode with the mode select register (PMDxMODESEL) of the motor control circuit (PMD).
- When the vector engine is used, the AD converter selects the program (trigger enable, AIN selection, and the result register selection) from the motor control circuit (PMD) according to the synchronous trigger.

18.2 Configuration

Figure 18-2 shows the configuration of the vector engine.

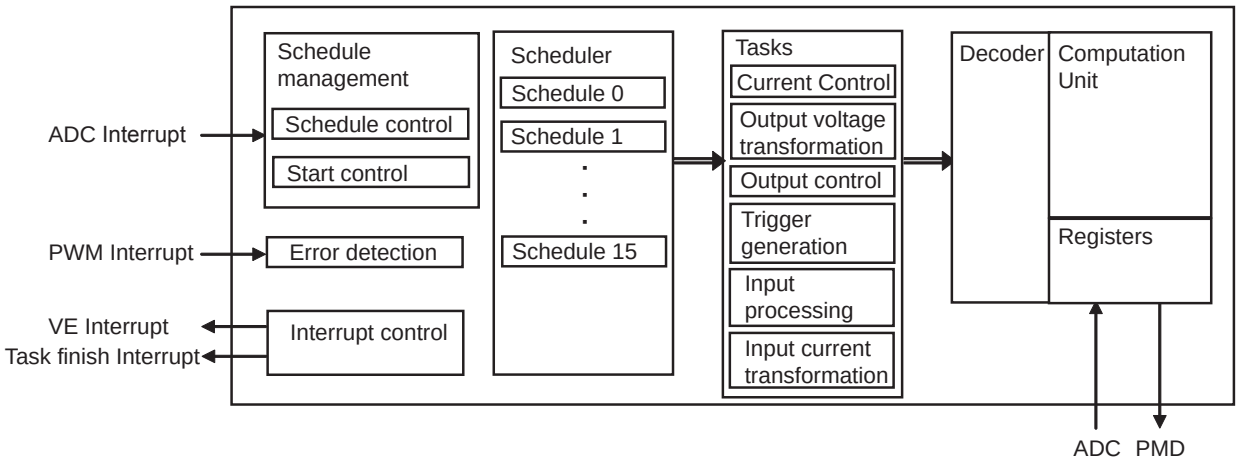


Figure 18-2 Configuration of the vector engine

18.2.1 Interaction between the Vector Engine, Motor Control Circuit, and AD Converter

As shown in Figure 18-3 and Figure 18-4, the vector engine allows to swap data between the motor control circuit (PMD) and AD converter (ADC) directly.

When the PMDxCMPU, PMDxCMPV, PMDxCMPW, PMDxMDOUT, PMDxTRGCMP0, PMDxTRGCMP1 and PMDxTRGSEL registers of the motor control circuit (PMD) is set to the VE mode with the PMDxMODESEL register, these registers are switched to VExCMPU, VExCMPV, VExCMPW, VExOUTCR, VExTRGCMP0, VExTRGCMP1 and VExTRGSEL of the vector engine registers. In this case, these vector engine registers cannot be controlled via the corresponding registers of the motor control circuit (PMD) by the CPU. These registers can be written by the vector engine. There are no read/write restrictions for other PMD registers.

The vector engine can read the value of the conversion result registers (ADxREG0, ADxREG1, ADxREG2, ADxREG3) of the AD converter (ADC) in the input process task. It can also read the conversion result from the two units ADCs and supports 2-phase current synchronous sampling. When the conversion result is read, phase information programed according to synchronous triggers from the PMD should be read.

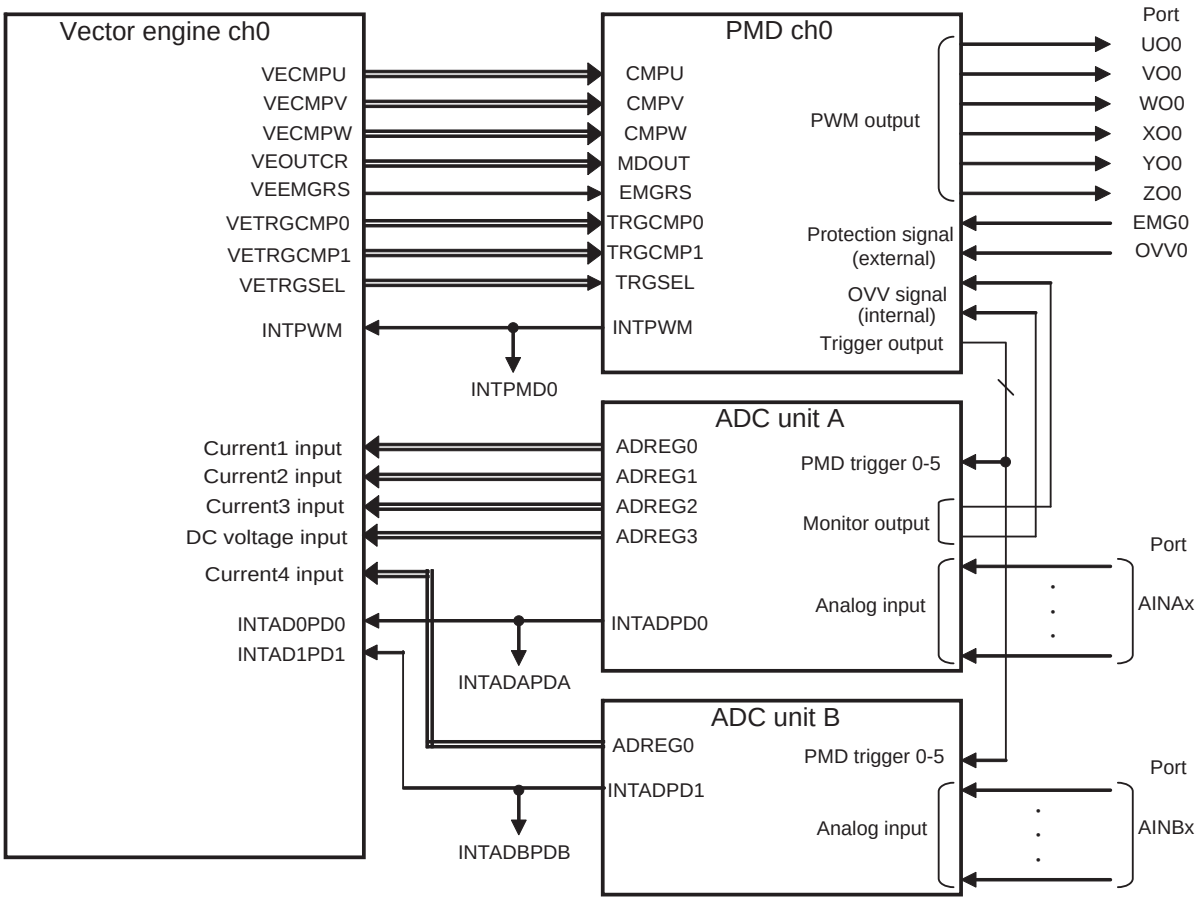


Figure 18-3 Block diagram of interaction between the vector engine, PMD and ADC (Channel 0)

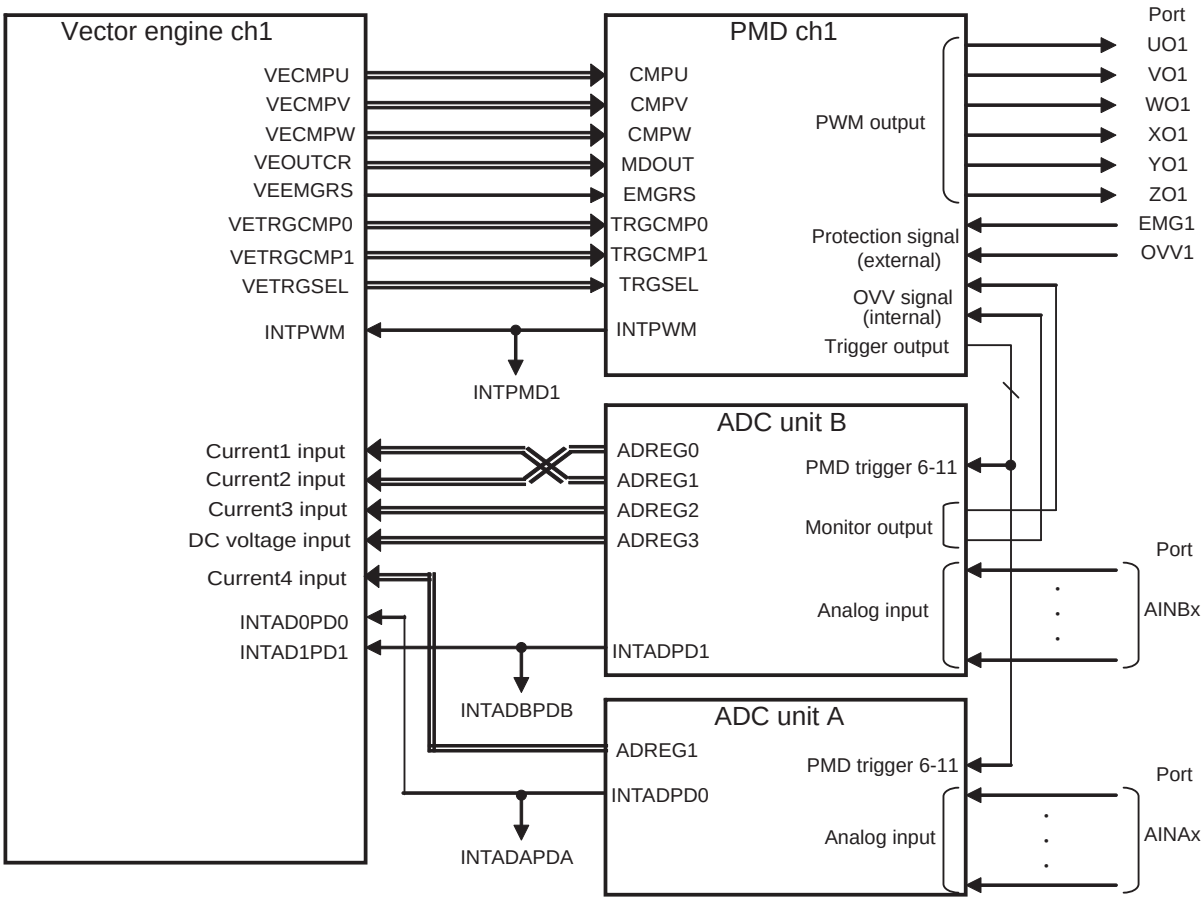


Figure 18-4 Block diagram of interaction between the vector engine, PMD and ADC (Channel 1)

18.3 Registers

The vector engine has the VE control registers and the dedicated registers.

- VE control registers

  The vector engine control registers and temporary registers.
- Dedicated registers

  Computation data registers and computation control registers.

18.3.1 List of Registers

VE control register

Register name			Address
Vector engine operation enable/disable	VExEN	R/W	0x0000
CPU start trigger selection	VExCPURUNTRG	W	0x0004
Task selection	VExTASKAPP	R/W	0x0008
Operation schedule selection	VExACTSCH	R/W	0x000C
Operation schedule repeat number selection	VExREPTIME	R/W	0x0010
Start trigger mode setting	VExTRGMODE	R/W	0x0014
Error interrupt enable/disable setting	VExERRINTEN	R/W	0x0018
Vector engine forcible termination	VExCOMPEND	W	0x001C
Error detection	VExERRDET	R	0x0020
Ongoing schedule flag/ongoing task	VExSCHTASKRUN	R	0x0024
Reserved	-	R	0x0028
Temporary 0	VExTMPREG0	R/W	0x002C
Temporary 1	VExTMPREG1	R/W	0x0030
Temporary 2	VExTMPREG2	R/W	0x0034
Temporary 3	VExTMPREG3	R/W	0x0038
Temporary 4	VExTMPREG4	R/W	0x003C
Temporary 5	VExTMPREG5	R/W	0x0040
Reserved	-	R	0x0214

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

Dedicated registers

Register name			Address
Abnormal determination result store	VExMCTLF	R/W	0x0044
Task control mode	VExMODE	R/W	0x0048
Flow control	VExFMODE	R/W	0x004C
PWM period rate (PWM period [s] × maximum speed (Note 1) $\times 2^{16}$) setting	VExTPWM	R/W	0x0050
Rotational speed (speed [Hz] ÷ maximum speed (Note 1) $\times 2^{15}$) setting	VExOMEGA	R/W	0x0054
Motor phase (motor phase [deg]/360 × 2^{16}) setting	VExTHETA	R/W	0x0058
d-axis reference value (current [A] ÷ maximum current (Note 2) $\times 2^{15}$) setting	VExIDREF	R/W	0x005C
q-axis reference value (current [A] ÷ maximum current (Note 2) $\times 2^{15}$) setting	VExIQREF	R/W	0x0060
d-axis voltage (voltage [V] ÷ maximum voltage (Note 3) $\times 2^{31}$) setting	VExVD	R/W	0x0064
q-axis voltage (voltage [V] ÷ maximum voltage (Note 3) $\times 2^{31}$) setting	VExVQ	R/W	0x0068

Dedicated registers

Register name			Address
Integral coefficient control setting for PI control of d-axis current	VExCIDKI	R/W	0x006C
Proportional coefficient setting for PI control of d-axis current	VExCIDKP	R/W	0x0070
Integral coefficient setting for PI control of q-axis current	VExCIQKI	R/W	0x0074
Proportional coefficient setting for PI control of q-axis current	VExCIQKP	R/W	0x0078
Integral component of d-axis voltage store (Upper 32 bits of VDI)	VExVDIH	R/W	0x007C
Integral component of d-axis voltage store (Lower 32 bits of VDI)	VExVDILH	R/W	0x0080
Integral component of q-axis voltage store (Upper 32 bits of VQI)	VExVQIH	R/W	0x0084
Integral component of q-axis voltage store (Lower 32 bits of VQI)	VExVQILH	R/W	0x0088
Switching speed of the PWM	VExFPWMCHG	R/W	0x008C
PWM period setting (set an identical value with the PWM cycle setting of the PMD).	VExMDPRD	R/W	0x0090
Minimum pulse width setting	VExMINPLS	R/W	0x0094
Synchronous trigger correction value setting	VExTRGCRC	R/W	0x0098
DC supply voltage (voltage [V] ÷ maximum voltage (Note 3) $\times 2^{15}$)	VExVDCL	R/W	0x009C
Cosine value at THETA for output conversion (Q15 data)	VExCOS	R/W	0x00A0
Sine value at THETA for output conversion (Q15 data)	VExSIN	R/W	0x00A4
Previous cosine value for input processing (Q15 data)	VExCOSM	R/W	0x00A8
Previous sine value for input processing (Q15 data)	VExSINM	R/W	0x00AC
Sector information	VExSECTOR	R/W	0x00B0
Previous sector information	VExSECTORM	R/W	0x00B4
a-phase zero-current (Note 4)	VExIAO	R/W	0x00B8
b-phase zero-current (Note 4)	VExIBO	R/W	0x00BC
c-phase zero-current (Note 4)	VExICO	R/W	0x00C0
AD conversion result of a-phase current (Note 4)	VExIAADC	R/W	0x00C4
AD conversion result of b-phase current (Note 4)	VExIBADC	R/W	0x00C8
AD conversion result of c-phase current (Note 4)	VExICADC	R/W	0x00CC
Supply voltage (voltage [V] ÷ maximum voltage (Note 3) $\times 2^{15}$)	VExVDC	R/W	0x00D0
d-axis current (current [A] ÷ maximum current (Note 2) $\times 2^{31}$)	VExID	R/W	0x00D4
q-axis current (current [A] ÷ maximum current (Note 2) $\times 2^{31}$)	VExIQ	R/W	0x00D8
Reserved	-	R/W	0x0174
AD conversion time setting	VExTADC	R/W	0x0178
PMD control: U-phase PWM pulse width setting	VExCMPU	R/W	0x017C
PMD control: V-phase PWM pulse width setting	VExCMPV	R/W	0x0180
PMD control: W-phase PWM pulse width setting	VExCMPÇv	R/W	0x0184
PMD control: 6-phase output control	VExOUTCR	R/W	0x0188
PMD control: Trigger timing setting (TRGCMP0)	VExTRGCMP0	R/W	0x018C
PMD control: Trigger timing setting (TRGCMP1)	VExTRGCMP1	R/W	0x0190
PMD control: Synchronous trigger selection	VExTRGSEL	R/W	0x0194
PMD control: EMG protection release	VExEMGRS	W	0x0198
PI control output limitation	VExPIOLIM	R/W	0x01BC
PI control d-axis coefficient range setting	VExCIDKG	R/W	0x01C0
PI control q-axis coefficient range setting	VExCIQKG	R/W	0x01C4
Voltage scalar limitation	VExVSLIM	R/W	0x01C8
Voltage scalar	VExVDQ	R/W	0x01CC
Declination angle	VExDELTA	R/W	0x01D0
Motor interlinkage magnetic flux	VExCPHI	R/W	0x01D4
Motor q-axis inductance	VExCLD	R/W	0x01D8
Motor d-axis inductance	VExCLQ	R/W	0x01DC
Motor resistance value	VExCR	R/W	0x01E0
Motor magnetic flux range setting	VExCPHIG	R/W	0x01E4

Dedicated registers

Register name			Address
Motor inductance range setting	VExCLG	R/W	0x01E8
Motor resistance range setting	VExCRG	R/W	0x01EC
Non-interference control for d-axis voltage	VExVDE	R/W	0x01F0
Non-interference control for q-axis voltage	VExVQE	R/W	0x01F4
Amount of dead time compensation	VExDTC	R/W	0x01F8
Hysteresis width for current discrimination	VExHYS	R/W	0x01FC
Dead time compensation control/status	VExDTCS	R/W	0x0200
PWM upper-limit setting	VExPWMMAX	R/W	0x0204
PWM lower-limit setting	VExPWMMIN	R/W	0x0208
Clipped phase value setting	VExTHTCLP	R/W	0x020C

Note 1: Do not access to the addresses described as "Reserved".
Note 2: The maximum speed: The number of controllable or operable maximum rotation [Hz]
Note 3: The maximum current: Variation ([A] × 2^{11}) of phase current when the result of AD conversion changes 1 LSB.
Note 4: The maximum voltage: Variation ([V] × 2^{12}) of supply voltage (VDC) when the result of AD conversion changes.
Note 5: The result of AD conversion is stored in the upper 12 bits of the 16-bit register.

18.3.2 VE Control Registers

18.3.2.1 VExEN (Vector Engine Operation 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	Selects clock operation in IDLE mode. 0: Stop 1: Operation
0	VEEN	R/W	Selects the vector engine operation. 0: Disabled 1: Enabled

Note: When the Vector Engine is disabled (VExEN = "0"), access to other registers of the vector engine is not allowed.

18.3.2.2 VExCPURUNTRG (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	-	-	-	-	-	-	-	VCPURT
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-1	−	R	Read as "0".
0	VCPURT	W	Starts the vector engine by software. 0: − 1: Starts operation Operation starts from the task configured to VExTASKAPP<VTASKA>. Specify VExTASKAPP, VExACTSCH, and VExREPTIME before starting operation.

Note 1: When "1" is written to this bit, it is cleared on the next cycle. This bit always read as "0".

Note 2: If new schedules and the tasks start operation while another schedule is being executed, the vector engine should be terminated with the VExCOMPEND register before it is started with the VExCPURUNTRG register again.

18.3.2.3 VExTASKAPP(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	-	-	-	-	VITASK			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	VTASK			
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-12	−	R	Read as "0".
11-8	VITASK[3:0]	R/W	Selects the timing at which the task completion interrupt occurs. 0x0: Output control 1 (Task number 0) 0x1: Trigger generation (Task number 1) 0x2: Input process 1 (Task number 2) 0x3: Input phase transformation (Task number 3) 0x4: Input coordinate axis transformation (Task number 4) 0x5: Current control (Task number 5) 0x6: SIN/COS calculation (Task number 6) 0x7: Output coordinate axis transformation (Task number 7) 0x8: Output phase transformation 1[SVM] (Task number 8) 0x9: Output control 2 (Task number 9) 0xA: Input process 2 (Task number 10) 0xB: Output phase transformation 2[I-Clarke] (Task number 11) 0xC: ATAN2 (Task number 12) 0xD: SQRT (Task number 13) 0xE,0xF: Reserved
7-4	−	R	Read as "0".
3-0	VTASK[3:0]	R/W	Selects the start task. 0x0: Output control 1 (Task number 0) 0x1: Trigger generation (Task number 1) 0x2: Input process 1 (Task number 2) 0x3: Input phase transformation (Task number 3) 0x4: Input coordinate axis transformation (Task number 4) 0x5: Current control (Task number 5) 0x6: SIN/COS calculation (Task number 6) 0x7: Output coordinate axis transformation (Task number 7) 0x8: Output phase transformation 1[SVM] (Task number 8) 0x9: Output control 2 (Task number 9) 0xA: Input process 2 (Task number 10) 0xB: Output phase transformation 2[I-Clarke] (Task number 11) 0xC: ATAN2 (Task number 12) 0xD: SQRT (Task number 13) 0xE,0xF: Reserved Selects the task that is started by software.

Note: Only those tasks that are included in the schedules can be specified.

18.3.2.4 VExACTSCH (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	-	-	-	-	VACT			
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-4	−	R	Read as "0".
3-0	VACT[3:0]	R/W	Selects individual task execution or an operation schedule. 0x0: Executes individual task. 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 For details, refer to "Table 18-4 Execution task in each schedule".

18.3.2.5 VExREPTIME (Operation Schedule Repeat Number 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	-	-	-	-	VREP			
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-4	−	R	Read as "0".
3-0	VREP[3:0]	R/W	Selects the number of repeat of the operation schedule. 0: No schedule operation is performed. 1-15: Repeats the specified number of times of the schedule operation.

Note: When this bit is set to "0", do not start schedule operation.

18.3.2.6 VExTRGMODE (Start Trigger 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	-	-	-	-	-	-	VTRG	
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-2	−	R	Read as "0".
1-0	VTRG[1:0]	R/W	Selects the input process start condition using an AD conversion completion interrupt. 00: An interrupt is ignored. 01: Input process is started by a INTADAPDA (ADCA PMD0 trigger synchronous conversion completion) interrupt. 10: Input process is started by a INTADBPDB (ADCB PMD1 trigger synchronous conversion completion) interrupt 11: Reserved

18.3.2.7 VExERRINTEN (Error Interrupt Enable/Disable 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	-	-	-	-	-	INTTEN	-	VERREN
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-3	−	R	Read as "0".
2	INTTEN	R/W	Controls a task completion interrupt. 0: Disabled 1: Enabled
1	−	R	Read as "0".
0	VERREN	R/W	Controls an interrupt at error detection. 0: Disabled 1: Enabled If a PWM interrupt is detected when an operation schedule is being executed (an activation trigger wait is not included), "1" is set as an error flag.

18.3.2.8 VExCOMPEND (VE Forcible Termination Register)

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

Bit	Bit symbol	Type	Function
31-1	−	R	Read as "0".
0	VCEND	W	Ongoing schedule is forcibly terminated. 0: − 1: Stop If "1" is written to this bit, this bit is cleared on next cycle. Always read as "0".

18.3.2.9 VExERRDET (Error Detection Register)

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

Bit	Bit symbol	Type	Function
31-1	−	R	Read as "0".
0	VERRD	R	This bit indicates the error flag. 0: An error is not detected. 1: An error is detected. When an operation schedule is being executed (excluding a start trigger wait), if a PWM interrupt is detected, "1" is set to this bit. This bit is cleared upon read.

18.3.2.10 VExSCHTASKRUN (Schedule Operation Status/Ongoing Task Number Register)

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

Bit	Bit symbol	Type	Function
31-5	−	R	Read as "0".
4-1	VRTASK[3:0]	R	Indicates the ongoing task number . 0x0: Output control 1 (Task number 0) 0x1: Trigger generation (Task number 1) 0x2: Input process 1 (Task number 2) 0x3: Input phase transformation (Task number 3) 0x4: Input coordinate axis transformation (Task number 4) 0x5: Current control (Task number 5) 0x6: SIN/COS calculation (Task number 6) 0x7: Output coordinate axis transformation (Task number 7) 0x8: Output phase transformation 1 [SVM] (Task number 8) 0x9: Output control 2 (Task number 9) 0xA: Input process2 (Task number 10) 0xB: Output phase transformation2[I-Clarke] (Task number 11) 0xC: ATAN2 (Task number 12) 0xD: SQRT (Task number 13) 0xE: Reserved 0xF: Reserved
0	VRSCH	R	Indicates schedule execution status. 0: Stop 1: Being executed

18.3.2.11 VExTMPREG0 to 5 (Temporary Registers)

VExTMPREG0

	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

VExTMPREG1

	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

VExTMPREG2

	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

VExTMPREG3

	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

VExTMPREG4

	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

VExTMPREG5

	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

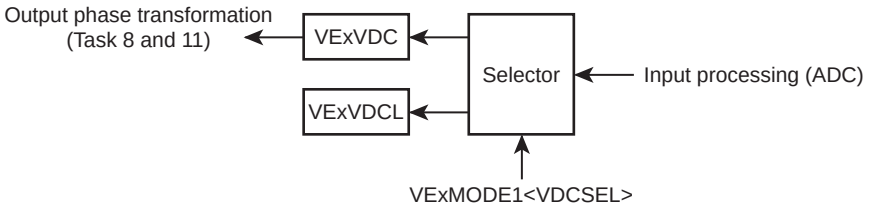
18.3.3 Dedicated Registers

18.3.3.1 VExMODE (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	IPDEN	PMDDTCEN	PWMFLEN	PWMBLEN	NICEN	T5ECEN	AWUMD	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	CLPEN	ATANMD		VDCSEL	OCRMD		ZIEN	PVIEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15	IPDEN	R/W	Disables/enables current polarity determination for Task 2 and Task 10. 0: Disabled 1: Enabled Refer to "18.4.2.6 Input Process".
14	PMDDTCEN	R/W	Setting that corresponds to the dead time correction control of the PMD when the dead time compensation for Task 0 and Task 9 is performed. 0: Dead time correction of the PMD is disabled. 1: Dead time correction of the PMD is enabled. Refer to "18.4.2.4 Output Control ".
13	PWMFLEN	R/W	Disables/enables a duty of 100% setting when the upper-limit for Task 0 and Task 9 is set. 0: Duty of 100% setting is disabled. 1: Duty of 100% setting is enabled. Refer to "18.4.2.4 Output Control ".
12	PWMBLEN	R/W	Disables/enables a duty of 0% setting when the lower-limit for Task 0 and Task 9 is set. 0: Duty of 0% setting is disabled. 1: Duty of 0% setting is enabled. Refer to "18.4.2.4 Output Control ".
11	NICEN	R/W	Disables/enables Non-interference control for Task 5. 0: Disabled 1: Enabled Refer to "18.4.2.1 Current Control (Task 5)".
10	T5ECEN	R/W	Disables/enables expansion control (Non-interference control and voltage scalar limitation) for Task 5. 0: Disabled 1: Enabled Refer to "18.4.2.1 Current Control (Task 5)".
9-8	AWUMD[1:0]	R/W	Specifies anti-windup (AWU) control for Task 5 when PI control output limitation is performed. 00: AWU control is disabled. 01: Substitutes the result from amount of limitation ÷ 4 into the integral term. 10: Substitutes the result from amount of limitation ÷ 2 into the integral term. 11: Substitutes the amount of limitation into the integral term. Refer to "18.4.2.1 Current Control (Task 5)".
7	CLPEN	R/W	Disables/enables phase clipping control for Task 6 when phase interpolation is performed. 0: Clipping is disabled. 1: Clipping is enabled. Refer to "18.4.2.2 SIN/COS calculation (Task 6)".

Bit	Bit symbol	Type	Function
6-5	ATANMD[1:0]	R/W	Specifies ATAN calculation control for Task 4. 0x: Calculation is disabled. 10: Calculates the declination angle on d-q coordinate of current vector. 11: Calculates the declination angle on d-q coordinate of induced voltage vector. Refer to "18.4.2.7 Input Current Transformation (Phase Transformation/Coordinate Axis Transformation)".
4	VDCSEL	R/W	Selects the supply voltage store register for Task 2 and 10. 0: Stored in VExVDC. 1: Stored in VExVDCL. Refer to "Figure 18-5 VExVDC/VExVDCL store register".
3-2	OCRMD[1:0]	R/W	Specifies output control for Task 0 and Task 9. 00: Output is disabled. 01: Output is enabled. 10: Short circuit brake (Outputs are OFF in the upper-phase; outputs are ON in the lower-phase.) 11: EMG recovery (Outputs are OFF.) Refer to "18.4.2.4 Output Control ".
1	ZIEN	R/W	Specifies zero-current detection control for Task 2. 0: Normal current detection 1: Zero-current detection Refer to "18.4.2.6 Input Process".
0	PVIEN	R/W	Disables/enables phase interpolation control for Task 6. 0: Disabled 1: Enabled Refer to "18.4.2.2 SIN/COS calculation (Task 6)".



Note) When a power supply voltage controlled by VExVDC register is corrected, select VExVDCL register as a storage location, and set a corrected value in VExDVC register.

Figure 18-5 VExVDC/VExVDCL store register

18.3.3.2 VExFMODE (Flow Control Register)

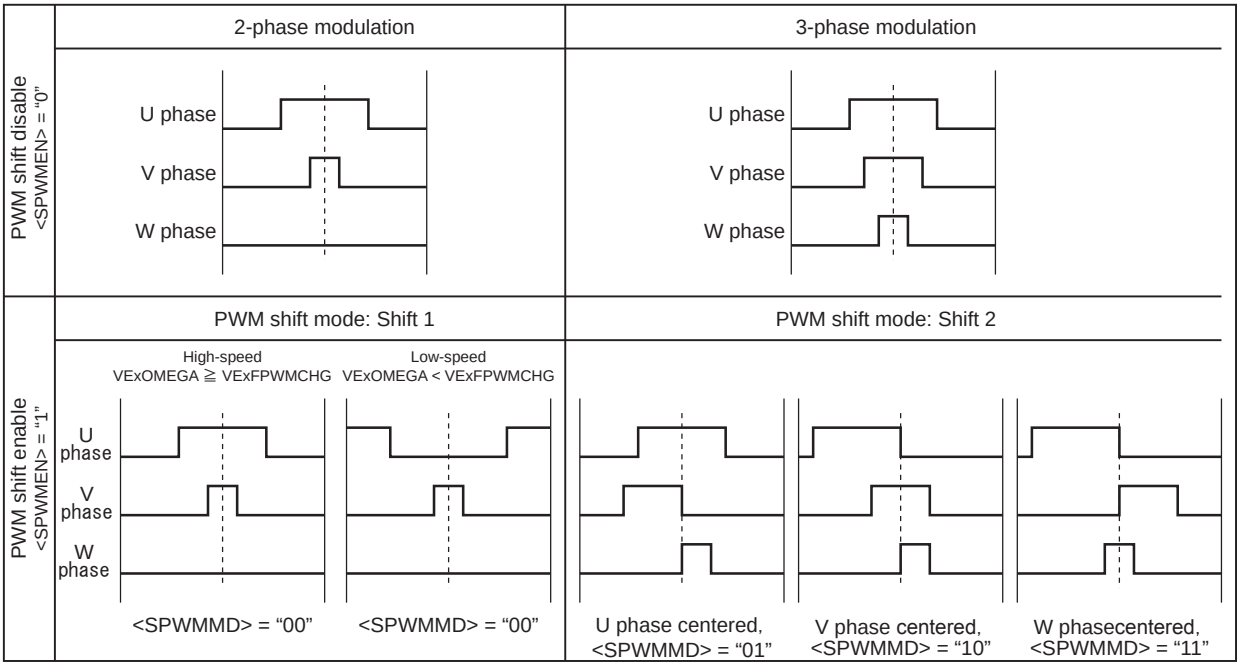
	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	SPWMMD		SADCEN	PHCVDIS	VSLIMMD		MREGDIS	CRCEN
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ICPLMD	IBPLMD	IAPLMD	-	IDMODE		SPWMEN	C2PEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-14	SPWMMD[1:0]	R/W	Selects a PWM shift mode. 00: Shift 1 01: Shift 2 (U-phase Normal PWM) 10: Shift 2 (V-phase Normal PWM) 11: Shift 2 (W-phase Normal PWM) For details, refer to "Figure 18-6 Relationship between shift setting and PWM waves". Note) Shift 2 cannot be selected when output control 1 (Task 0) is being executed (invalid).
13	SADCEN	R/W	Disables/enables synchronous current sampling control using two units of the ADC. 0: Synchronous sampling is disabled. 1: Synchronous sampling is enabled.
12	PHCVDIS	R/W	Disables/enables phase transformation. 0: 2-3 phase transformation is enabled. (3-phase AC output) 1: 2-3 phase transformation is disabled. (2-phase AC output) Note) Phase transformation cannot be disabled when space vector modulation (Task 8) is running (invalid).
11-10	VSLIMMD [1:0]	R/W	Controls voltage scalar limitation of Task 5. 00: Scalar limitation is disabled. (The limitation of each axis is enabled.) 01: Limits the voltage on the d-axis. 10: Limits the voltage on the q-axis. 11: dq proportional limitation
9	MREGDIS	R/W	Stores the previous value of SIN/COS/SECTOR 0: Enabled 1: Disabled When this bit is disabled, VExSINM = VExSIN, VExCOSM = VExCOS, and VExSECTORM = VExSECTOR are set.
8	CRCEN	R/W	Disables/enables trigger correction. 0: Disabled 1: Enabled Note: This bit is enabled when trigger generation (Task 1) is being executed; when PWM shift is disabled in 1-shunt current detection mode; or when Shift 1 is selected.
7	ICPLMD	R/W	Selects the current direction (Ic) of Task 10. 0: Shunt mode (Ic = VExICO − VExICADC) 1: Sensor mode (Ic = VExICADC − VExICO)
6	IBPLMD	R/W	Selects the current direction (Ib) of Task 10. 0: Shunt mode (Ib = VExIBO − VExIBADC) 1: Sensor mode (Ib = VExIBADC − VExIBO)
5	IAPLMD	R/W	Selects the current direction (Ia) of Task 10. 0: Shunt mode (Ia = VExIAO − VExIAADC) 1: Sensor mode (Ia = VExIAADC − VExIAO)

Bit	Bit symbol	Type	Function
4	−	R/W	Write as "0".
3-2	IDMODE[1:0]	R/W	Selects current detection mode. 00: 3-shunt (Note 1) 01: 2-sensor (Note 2) 10: 1-shunt (PMD TRG up counter (Note 3, Note 4)) 11: 1-shunt (PMD TRG down counter (Note 3, Note 4)) Note 1: 3-phase current detection is used in input process 2 (Task 10). Note 2: 2-phase current detection is used in input process 2 (Task 10). Note 4) PWM Shift 2 should be used when output control 2 (Task 9) and input process 2 (Task 10) is being executed.
1	SPWMEN	R/W	Disables/enables PWM shift. 0: Disabled 1: Enabled Note: Output control 1 (Task 0) and input process 1 (Task 2) are available only in Shift 1 mode. Output control 2 (Task 9) and input process 2 (Task 10) are available only in Shift 2 mode.
0	C2PEN	R/W	Selects the modulation mode. 0: 3-phase modulation 1: 2-phase modulation

Note 3) The setting of PMDTRG when 1-shunt mode is used.

VExFMODE <IDMODE[1:0]>	PMDxTRGCR <TRG0MD[2:0]>	PMDxTRGCR <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 18-6 Relationship between shift setting and PWM waves

18.3.3.3 VExTPWM (PWM Cycle Rate 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	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	Sets the PWM cycle rate and the unit of integration in phase interpolation (16-bit fixed-point data: 0.0 to 1.0). 0x0000 to 0xFFFF (PWM cycle[s] × Max_Hz × 2^{16}) This indicates the ratio between PWM cycle and the maximum rotational speed. (Max_Hz: Maximum rotational speed) This bit is used for SIN/COS calculation (Task 6) when phase interpolation is enabled.

18.3.3.4 VExOMEGA (Rotational 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	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	Sets a rotational speed (16-bit fixed-point data: −1.0 to 1.0). 0x8000 to 0x7FFF (Rotational speed [Hz] ÷ Max_Hz × 2^{15}) (Max_Hz: Maximum rotational speed) This bit is used for SIN/COS calculation (Task 6) when phase interpolation is enabled. This bit is used for output control 1 (Task 0) when PWM Shift 1 is selected for 1-shunt current detection. This bit is used for current control (Task 5) and input coordinate axis transformation (Task 4) in motor voltage equation.

18.3.3.5 VExTHETA (Motor Phase 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	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	Sets the phase (16-bit fixed-point data: 0.0 to 1.0). Setting value: Phase [deg] ÷ 360 × 2^{16} This bit is used for SIN/COS calculation (Task 6) when phase interpolation is enabled. This bit is updated when SIN/COS calculation (Task 6) is executed while phase interpolation is enabled.

18.3.3.6 VExCOS/VExSIN/VExCOSM/VExSINM (SIN/COS Register)

VExCOS

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the cosine value on the THETA (16-bit fixed-point data: −1.0 to 1.0). 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for output coordinate axis transformation (Task 7).

VExSIN

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the sine value on the THETA (16-bit fixed-point data: −1.0 to 1.0). 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for output coordinate axis transformation (Task 7).

VExCOSM

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Stores the previous value of the VExCOS register. 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for input coordinate axis transformation (Task 4).

VExSINM

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Stores the previous value of the VExSIN register. 0x8000 to 0x7FFF This bit is updated when SIN/COS calculation (Task 6) is executed. This bit is used for input coordinate axis transformation (Task 4).

18.3.3.7 VExIDREF/VExIQREF (d-axis/q-axis Current Reference Setting Registers)

VExIDREF

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the reference value of d-axis current (16-bit fixed-point data: −1.0 to 1.0). 0x8000 to 0x7FFF (d-axis current reference [A] ÷ Max_I × 2^{15}) Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11} This bit is used for current control (Task 5).

VExIQREF

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the reference value of q-axis current (16-bit fixed-point data: −1.0 to 1.0). 0x8000 to 0x7FFF (q-axis current reference [A] ÷ Max_I × 2^{15}) Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11} This bit is used for current control (Task 5).

18.3.3.8 VExVD/VExVQ (d-axis/q-axis Voltage Setting Register)

VExVD

	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	Sets the value of d-axis voltage (32-bit fixed-point data: −1.0 to 1.0). 0x8000_0000 to 0x7FFF_FFFF: d-axis voltage[V] ÷ Max_V × 2^{31} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} This bit is updated when current control (Task 5) is executed. This bit is used for output coordinate axis transformation (Task 7). This bit is used for input coordinate axis transformation (Task 4) in motor voltage equation.

VExVQ

	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[V] ÷ Max_V × 2^{31} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} This bit is updated when current control (Task 5) is executed. This bit is used on output coordinate axis transformation (Task 7). This bit is used on input coordinate axis transformation (Task 4) for motor voltage equation.

18.3.3.9 VExCIDKI/VExCIDKP/VExVCIQKI/VExCIQKP (PI Control Coefficient Registers)

VExCIDKI

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the PI control integral coefficient for d-axis (0x8000 to 0x7FFF).

VExCIDKP

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the PI control proportional coefficient for d-axis (0x8000 to 0x7FFF).

VExVCIQKI

	31	30	29	28	27	26	25	24
Bit symbol	-							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the PI control integral coefficient for q-axis (0x8000 to 0x7FFF).

VExCIQKP

	31	30	29	28	27	26	25	24
Bit symbol	-							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the PI control probational coefficient for q-axis (0x8000 to 0x7FFF).

18.3.3.10 VExVDIH/VExVDILH/VExVQIH/VExVQILH (PI Control Integral component Store Registers)

VExVDIH

	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	VDIH							
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-0	VDIH[31:0]	R/W	Stores the upper 32 bits of the integral component (VDI) for d-axis on PI control.

VExVDILH

	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	Stores the lower 16 bits of the integral component (VDI) for d-axis on PI control.
15-0	−	R	Read as "0".

Note 1: The VDI data is 64-bit fixed-point data (63 fractional bits from −1.0 to 1.0).

Note 2: The VDI data consists of 48 bits.

VExVQIH

	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	Stores the upper 32 bits of the integral component (VQI) for q-axis on PI control.

VExVQILH

	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	Stores the lower 16 bits of the integral component (VQI) for q-axis on PI control.
15-0	−	R	Read as "0".

Note 1: The VQI data is 64-bit fixed-point data (63 fractional bits from -1.0 to 1.0).

Note 2: The VQI data consists of 48 bits.

18.3.3.11 VExMCTLF(Abnormal Determination Result Store Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	PWMOVF	VSOVF	PIQOVF	PIDOVF
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
31-16	−	R	Read as "0".
15-12	−	R/W	Write as "0".
11	PWMOVF	R/W	Indicates the excess flag for PWM output limitation. 0: All 3-phase PWM outputs are PWMMIN or more, and PWMMAX or less. 1: Any 3-phase PWM outputs are less than PWMMIN, or over PWMMAX. This bit is updated when output controls (Task 0 and Task 9) are executed.
10	VSOVF	R/W	Indicates the excess flag for voltage scalar limitation. 0: Voltage scalar ≤ VSLIM 1: Voltage scalar > VSLIM This bit is updated when current control (Task 5) is executed.
9	PIQOVF	R/W	Indicates the excess flag for q-axis output limitation on PI control. 0: q-axis output on PI control ≤ PIOLIM 1: q-axis output on PI control > PIOLIM This bit is updated when current control (Task 5) is executed.
8	PIDOVF	R/W	Indicates the excess flag for d-axis output limitation on PI control. 0: d-axis output on PI control ≤ PIOLIM 1: d-axis output on PI control > PIOLIM This bit is updated when current control (Task 5) is executed.
7-6	−	R/W	Write as "0".
5	PLSLFM	R/W	Indicates the previous value of the <PLSLF> register. This bit is updated when output controls (Task 0 and Task 9) are executed.
4	PLSLF	R/W	Sets the smallest pulse flag. When output control 1 (Task 0) is executed and 1-shunt current detection is set, Difference of minimum pulse width < VExMINPLS<MINPLS>. If so, set "1". When output control 2 (Task 9) is executed, Minimum On-width or minimum Off-width < VExMINPLS<MINPLS>. If so, set "1".
3	−	R/W	Write as "0".
2	LVTF	R/W	Selects the low supply voltage flag. VExVDC<VDC> ≥ 1/128. If so, set "0". VExVDC<VDC> < 1/128. If so, set "1". This bit is updated when output phase transformations (Task 8 and 11) are executed.
1	LAVFM	R/W	Indicates the previous value of the <LAVF> register. This bit is updated when output control 1 (Task 0) is executed and PWM shift is enabled in 1-shunt current detection.

Bit	Bit symbol	Type	Function
0	LAVF	R/W	Selects the low-speed flag. When output control 1 (Task 0) is executed, and PWM shift is enabled in 1-shunt current detection. 0: High speed 1: Low speed VExOMEGA<OMEGA> ≥ VExFPWMCHG<FPWMCHG>. If so, set "0". VExOMEGA<OMEGA> < VExFPWMCHG<FPWMCHG>. If so, set "1".

18.3.3.12 VExFPWMCHG (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. 0x0000 to 0x7FFF (Switching speed [Hz] ÷ Max_Hz × 2^{15}) (Max_Hz: Maximum rotational speed [Hz]) This bit is used when output control 1 (Task 0) is executed and PWM Shift 1 is enabled in 1-shunt current detection.

18.3.3.13 VExMDPRD (PWM Cycle 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	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	Set the PWM cycle. Sets the same value as the PMDxMDPRD register of the PMD circuit. This bit is used for output controls (Task 0 and Task 9) and trigger generation (Task 1).

18.3.3.14 VExMINPLS (Minimum Pulse Width Difference 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	Sets the reference value of the minimum pulse width difference (the minimum value of duty of three-phase PWM (VExCMPU,VExCMPV, and VExCMPW)) when output control 1 (Task 0) is executed and PWM shift is enabled in the 1-shunt current detection. The calculation is as follows: Difference of pulse width [s] ÷ PWMcounter clock cycle [s] Sets the reference value of the minimum pulse width (the minimum value of duty of three phase PWM (VExCMPU,VExCMPV,and VExCMPW)) when output control 2 (Task 9) is executed. The calculation is as follows: Pulse width [s] ÷ PWM counter clock cycle [s]

18.3.3.15 VExSECTOR/VExSECTORM (Sector Information Register)

VExSECTOR

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	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	Indicates the sector information. Setting value: 0x0 to 0xB A rotational position when output is expressed in the sector that is divided into 12 sectors by 30 degrees. This bit is updated when output phase transformations (Task 8 and Task 11) are executed. This bit is used for output control 1 (Task 0).

VExSECTORM

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	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	Indicates the previous sector information. Setting value: 0x0 to 0xB This bit is used in input process. This bit is updated when output phase transformations (Task 8 and Task 11) are executed. This bit is used in input process 1 (Task 2).

18.3.3.16 VExIAO/VExIBO/VExICO (Zero Current Register)

VExIAO

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Stores the result of AD conversion for a-phase at zero current. (Stores the result of AD conversion for a-phase current when the motor is at stop.) This bit is updated in the input process 1 (Task 2) when zero-current detection mode is selected. The result of AD conversion for a-phase at zero current is stored in <IAO[15:4]> when the result of AD conversion is captured. <IAO[3:0]> stores "0".

VExIBO

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Stores the result of AD conversion for b-phase at zero current. (Stores the result of AD conversion for b-phase current when the motor is at stop.) This bit is updated in the input process 1 (Task 2) when zero-current detection mode is selected. The result of AD conversion for b-phase at zero current is stored in <IBO[15:4]> when the result of AD conversion is captured. <IBO[3:0]> stores "0".

VExICO

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Stores the result of AD conversion for c-phase at zero current. (Stores the result of AD conversion for c-phase current when the motor is at stop.) This bit is updated in the input process 1 (Task 2) when zero-current detection mode is selected. The result of AD conversion for c-phase at zero current is stored in <ICO[15:4]> when the result of AD conversion is captured. <ICO[3:0]> stores "0".

18.3.3.17 VExIAADC/VExIBADC/VExICADC (ADC Current Result Register)

VExIAADC

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 result of AD conversion for a-phase current (0x0000 to 0xFFFF). This bit is updated when input processes (Task 2 and Task 10) are executed. The result of AD conversion for a-phase is stored in <IAADC[15:4]> when the result of AD conversion is captured. <IAADC[3:0]> stores "0".

VExIBADC

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 result of AD conversion for b-phase current (0x0000 to 0xFFFF). This bit is updated when input processes (Tasks 2 and10) are executed. The result of AD conversion for b-phase is stored in <IBADC[15:4]> when the result of AD conversion is captured. <IBADC[3:0]> stores "0".

VExICADC

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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 result of AD conversion for c-phase current (0x0000 to 0xFFFF). This bit is updated when the input processes (Task 2 and Task 10) are executed. The result of AD conversion for c-phase is stored in <ICADC[15:4]> when the result of AD conversion is captured. <ICADC[3:0]> stores "0".

18.3.3.18 VExVDC/VExVDCL (Supply Voltage Register)

VEVDCx

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets a supply voltage (16-bit fixed-point data: 0 to 1.0). Setting value: 0x0000 to 0x7FFF The actual voltage value can be calculated as: the value of VDC × the value of Max_V ÷ 2^{15} (Max_V: (The amount of supply voltage variation when AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when VExMODE<VDCSEL> is set to "0" and input processes (Task 2 and Task 10) are executed. This bit is used for output phase transformation (Tasks 8 and 11).

VExVDCL

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets a supply voltage (16-bit fixed-point data: 0 to 1.0). Setting value: 0x0000 to 0x7FFF The actual voltage value can be calculated as: the value of VDCL × the value of Max_V ÷ 2^{15} (Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when VExMODE<VDCSEL> is set to "1" and input processes (Task 2 and Task 10) are executed.

18.3.3.19 VExID/VExIQ (d-axis/q-axis Current Register)

VExID

	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	Sets a d-axis current (32-bit fixed-point data: −1.0 to 1.0). Setting value: 0x8000_0000 to 0x7FFF_FFFF The actual voltage value can be calculated as: the value of ID × the value of Max_I ÷ 2^{31} (Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11}) This bit is updated when input coordinate axis transformation (Task 4) is executed. This bit is used for current control (Task 5).

VExIQ

	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	Sets a q-axis current (32-bit fixed-point data: −1.0 to 1.0). Setting value: 0x8000_0000 to 0x7FFF_FFFF The actual voltage value can be calculated as: the value of IQ × the value of Max_I ÷ 2^{31} (Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11}) This bit is updated when input coordinate axis transformation (Task 4) is executed. This bit is used for current control (Task 5).

18.3.3.20 VExTADC (ADC Conversion Time 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 an AD conversion time (Refer to Figure 18-7). 0x0000to0xFFFF: (AD conversion time [s] ÷ PWM counter clock cycle [s]) Performs forward correction for a trigger timing at the carrier peak when PWM Shift 1 output (Shift 1 is enabled and a low-speed flag is "1") is used in 1-shunt current detection mode. This bit is used in Task 0.

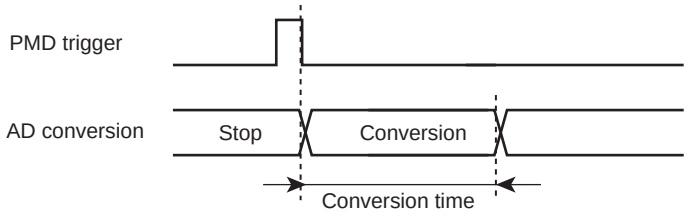


Figure 18-7 ADC conversion time

18.3.3.21 VExCMPU/ VExCMPV/ VExCMPW (PWM Duty Registers)

VExCMPU

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the U-phase PWM pulse width. Setting value: 0x0000 to 0xFFFF This bit is updated when output controls (Task 0 and Task 9) are executed. This bit is used for trigger generation (Task 1).

VExCMPV

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the V-phase PWM pulse width. Setting value: 0x0000 to 0xFFFF This bit is updated when output controls (Task 0 and Task 9) are executed. This bit is used for trigger generation (Task 1).

VExCMPW

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Sets the W-phase PWM pulse width. Setting value: 0x0000 to 0xFFFF This bit is updated when output controls (Task 0 and Task 9) are executed. This bit is used for trigger generation (Task 1).

18.3.3.22 VExOUTCR (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	Select W-phase PWM. 0: ON/OFF output 1: PWM output
7	VPWM	R/W	Selects V-phase PWM 0: ON/OFF output 1: PWM output
6	UPWM	R/W	Selects U-phase PWM 0: ON/OFF output 1: PWM output
5-4	WOC[1:0]	R/W	Selects W-phase output control. 00: WO OFF, ZO OFF (Note) 01: WO OFF, ZO ON 10: WO ON, ZO OFF 11: WO ON, ZO ON (Note) Both WO and ZO are ON when <WPWM> = 1.
3-2	VOC[1:0]	R/W	Selects V-phase output control. 00: VO OFF, YO OFF (Note) 01: VO OFF, YO ON 10: VO ON, YO OFF 11: VO ON, YO ON (Note) Both VO and YO are ON when <VPWM> = 1.
1-0	UOC[1:0]	R/W	Selects U-phase output control. 00: UO OFF, XO OFF (Note) 01: UO OFF, XO ON 10: UO ON, XO OFF 11: UO ON, XO ON (Note) Both UO and XO are ON when <UPWM> = 1.

Note: This setting is updated when output controls (Task 0 and Task 9) are executed.

Output control of U,V and W-phase of the PMD is shown below: (This table shows only those combinations that are used in the VE.)

Table 18-1 <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 18-2 <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 18-3 <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

18.3.3.23 VExTRGCRC (Synchronous Trigger Correction 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	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	Corrects the synchronous trigger timing. Setting value: Correction time [s] ÷ PWMcounter clock cycle [s] This bit is enabled only when PWM shift is disabled in the 1-shunt current detection or when Shift 1 is enabled. This bit is used to backward correct the trigger timing. This bit is used for trigger generation (Task 1).

18.3.3.24 VExTRGCMP0/VExTRGCMP1 (Trigger Timing Setting Register)

VExTRGCMP0

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Specifies the trigger timing for sampling ADC in synchronization with the PMD (PMD setting). 0x0000: Prohibited 0x0001 to (the value of VExMDPRD−1): Trigger timing The value of VExMDPRD to 0xFFFF: Prohibited This bit is enabled when one of the following PMD trigger modes is selected: A match of down counting with the timer; a match of up counting with the timer; or matches of up and down counting with the timer This bit is updated in trigger generation (Task 1) when PWM shift is disabled in the 1-shunt current detection or when Shift 1 is enabled.

VExTRGCMP1

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	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	Specifies the trigger timing for sampling ADC in synchronization with the PMD (PMD setting). 0x0000: Prohibited 0x0001 to (The value of VExMDPRD−1): Trigger timing The value of VExMDPRD to 0xFFFF: Prohibited This bit is enabled when one of the following PMD trigger modes is selected: A match of down counting with the timer; a match of up counting with the timer; or matches of up and down counting with the timer This bit is updated in trigger generation (Task 1) when PWM shift is disabled in the 1-shunt current detection or when Shift 1 is enabled.

18.3.3.25 VExTRGSEL (Synchronous 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	Specifies the synchronous trigger number that is output at the timing specified in VExTRGCMP0 (PMD setting). 0 to 5: Output trigger number 6 to 7: Prohibited This bit is enabled when trigger selection output (PMDxTRGMD<TRGOUT> = "1") is selected as PMD trigger output mode. This bit is updated to the value of VExSECTOR ÷ 2 in trigger generation (Task 1).

18.3.3.26 VExEMGRS (EMG protection 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	-	-	-	-	-	-	-	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	EMG protection release command (PMD setting) 0: - 1: EMG protection release If "1" is written to this bit, this bit is cleared on next cycle. Always read as "0". This bit is set to "1" when output control (Task 0 and Task 9) is executed in EMG recovery mode.

18.3.3.27 VExPIOLIM (PI Control Output Limitation Register)

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	PIOLIM[15:0]	R/W	PI control output limit value setting Setting value: 0x0000 to 0x7FFF This bit is used for current control (Task 5).

18.3.3.28 VExCIDKG/VExCIQKG (Coefficient Range Setting Register for d-axis/q-axis of PI Control)

VExCIDKG

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

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7-0	CIDKG[7:0]	R/W	Selects the coefficient range for d-axis of PI control. 0x00: 1/1 0x01: $1/2^4$ 0x02: $1/2^8$ 0x03: $1/2^{12}$ 0x04: $1/2^{16}$ 0x05 to 0xFF: Reserved This bit is used for current control (Task 5).

VExCIQKG

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

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7-0	CIQKG[7:0]	R/W	Selects the coefficient range for q-axis of PI control. 0x00: 1/1 0x01: $1/2^4$ 0x02: $1/2^8$ 0x03: $1/2^{12}$ 0x04: $1/2^{16}$ 0x05 to 0xFF: Reserved This bit is used for current control (Task 5).

18.3.3.29 VExVSLIM (Voltage Scalar Limitation Register)

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	VSLIM[15:0]	R/W	Sets the limitation value of voltage scalar for d-axis voltage (VExVD) and q-axis voltage (VExVQ). 0x0000 to 0x7FFF: Voltage [V] ÷ Max_V × 2^{15} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} Limitation of scalar voltage is not effective when 0x0000 is set. This bit is used for current control (Task 5) when expansion control is enabled.

18.3.3.30 VExVDQ (Voltage Scalar Register)

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	VDQ[15:0]	R/W	Sets value of voltage scalar for d-axis voltage (VExVD) and q-axis voltage (VExVQ), or sets the limitation value of axis for d-axis voltage (VExVD) and q-axis voltage (VExVQ) when direction scalar is limited. 0x0000 to 0x7FFF: Voltage [V] ÷ Max_V × 2^{15} Max_V: (The amount of supply voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12} This bit is used for current control (Task 5) when expansion control is enabled.

18.3.3.31 VExDELTA (Declination Angle Register)

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	DELTA[15:0]	R/W	Sets the declination angle on the d-q coordinate. Range: 0x8000 to 0x7FFF(−180 degrees to 180 degrees) The declination angle: [deg] ÷ 360 × 2^{16} This bit is updated when ATAN calculation is enabled on coordinate axis transformation (Task 4) or when voltage scalar limitation is enabled in current control (Task 5).

18.3.3.32 VExCPHI/VExCLD/VExCLQ/VExCR/VExCPHIG/VExCLG/VExCRG (Motor Constant Registers)

VExCPHI

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CPHI[15:0]	R/W	Sets the back electromotive force constant of the motor [V/rps] (interlinkage flux [Wb/s]) Setting value: 0x0000 to 0x7FFF The value of back electromotive force constant [V/rps] ÷ Max_V × Max_Hz × 2^{11} ÷ [CPHIG setting] This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

VExCLD

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CLD[15:0]	R/W	Sets the d-axis inductance of the motor constant. Setting value: 0x0000 to 0x7FFF The value of inductance [H] × Max_I ÷ Max_V × Max_Hz × 2π × 2^{11} ÷ [CLG setting] This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

VExCLQ

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CLQ[15:0]	R/W	Sets the q-axis inductance of the motor constant. Setting value: 0x0000 to 0x7FFF The value of inductance [H] × Max_I ÷ Max_V × Max_Hz × 2π × 2^{11} ÷ [CLG setting] This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

VExCR

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	CR[15:0]	R/W	Sets the value of resistor of the motor constant. Setting value: 0x0000 to 0x7FFF The value of resistor [Ω] × Max_I ÷ Max_V × 2^{11} ÷ [CRG setting] This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

VExCPHIG

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

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7-0	CPHIG[7:0]	R/W	Selects the range of magnetic flux of the motor constant. 0x00: 1/1 0x01: $1/2^4$ 0x02: $1/2^8$ 0x03: $1/2^{12}$ 0x04: $1/2^{16}$ 0x05 to 0xFF: Reserved This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

VExCLG

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

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7-0	CLG[7:0]	R/W	Selects the range of inductance of the motor constant. 0x00: 1/1 0x01: $1/2^4$ 0x02: $1/2^8$ 0x03: $1/2^{12}$ 0x04: $1/2^{16}$ 0x05 to 0xFF: Reserved This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

VExCRG

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

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7-0	CRG[7:0]	R/W	Selects the range of the resistor of the motor constant. 0x00: 1/1 (The CR register uses [31:16] on Q27 format.) 0x01: $1/2^4$ 0x02: $1/2^8$ 0x03: $1/2^{12}$ 0x04: $1/2^{16}$ 0x05 to 0xFF: Reserved This bit is used in reference axis transformation (Task 4) when the declination angle of induced voltage is calculated or when Non-interference control is enabled in current control (Task 5).

18.3.3.33 VExVDE/VExVQE(Non-interference Control Voltage Register for d-axis/q-axis)

VExVDE

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	VDE[15:0]	R/W	Indicates the value of d-axis calculation of non-interference (16-bit fixed-point data from -1.0 to 1.0). 0x8000 to 0x7FFF: Voltage [V] ÷ Max_V × 2^{15} (Max_V: (The amount of voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when Non-interference control is enabled in current control (Task 5).

VExVQE

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	VQE[15:0]	R/W	Indicates the value of q-axis calculation of non-interference (16-bit fixed-point data from -1.0 to 1.0). 0x8000 to 0x7FFF: Voltage [V] ÷ Max_V × 2^{15} (Max_V: (The amount of voltage variation when the AD conversion is changed by 1LSB [V]) × 2^{12}) This bit is updated when Non-interference control is enabled in current control (Task 5).

18.3.3.34 VExDTC (Dead Time Compensation Register)

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	DTC	R/W	Sets the amount of compensation on dead time time control. 0x0000 to 0xFFFF: Dead time [s] ÷ PWM cycle [s] × the value of VExMDPRD This bit is used when dead time compensation is enabled in output controls (Task 0 and Task 9).

18.3.3.35 VExHYS(Hysteresis width for current discrimination)

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	HYS[15:0]	R/W	Sets the current hysteresis width when current polarity is determined (16-bit fixed-point data from -1.0 to 1.0). 0x8000 to 0x7FFF: Hysteresis width [A] ÷ Max_I × 2^{15} (Max_I: (The amount of phase current variation when the AD conversion is changed by 1LSB [A]) × 2^{11}) This bit is used in input process (Task 2 and Task 10) when current polarity determination is enabled.

18.3.3.36 VExDTCS (Dead Time Compensation Control/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	-	-	-	-	-	ICSTS		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	IBSTS			-	IASTS		
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-11	−	R/W	Write as "0".
10-8	ICSTS[1:0]	R/W	Indicates status of the current (Ic) polarity determination,or selects dead time compensation control setting for CMPW. xx0: Polarity determination is undefined. Dead time compensation is not performed. x01: Positive current is determined. Adds the DTC value when dead time compensation is performed. x11: Negative current is determined. Subtracts the DTC value when dead time compensation is performed. This bit is updated in input process (Task 2 and Task 10) when current polarity determination is enabled. This bit is used in output controls (Task 0 and Task 9) when dead time compensation is enabled.
7	−	R/W	Write as "0".
6-4	IBSTS[1:0]	R/W	Indicates status of the current (Ib) polarity determination,or selects dead time compensation control setting for CMPV. xx0: Polarity determination is undefined. Dead time compensation is not performed. x01: Positive current is determined. Adds the DTC value when dead time compensation is performed. x11: Negative current is determined. Subtracts the DTC value when dead time compensation is performed. This bit is updated in input process (Task 2 and Task 10) when current polarity determination is enabled. This bit is used in output controls (Task 0 and Task 9) when dead time compensation is enabled.
3	−	R/W	Write as "0".
2-0	IASTS[1:0]	R/W	Indicates status of the current (Ia) polarity determination,or selects dead time compensation control setting for CMPU. xx0: Polarity determination is undefined. Dead time compensation is not performed. x01: Positive current is determined. Adds the DTC value when dead time compensation is performed. x11: Negative current is determined. Subtracts the DTC value when dead time compensation is performed. This bit is updated in input process (Task 2 and Task 10) when current polarity determination is enabled. This bit is used in output controls (Task 0 and Task 9) when dead time compensation is enabled.

18.3.3.37 VExPWMMAX/VExPWMMIN (PWM Output Limitation Register)

VExPWMMAX

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	PWMMAX[15:0]	R/W	Sets the upper limit value of PWM output. Setting value: 0x0000 to 0xFFFF This bit is used in output controls (Task 0 and Task 9).

VExPWMMIN

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

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	PWMMIN[15:0]	R/W	Sets the lower limit value of PWM output. Setting value: 0x0000 to 0xFFFF This bit is used in output controls (Task 0 and Task 9).

18.3.3.38 VExTHTCLP(Clipped phase 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	THTCLP							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	THTCLP							
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-16	−	R	Read as "0".
15-0	THTCLP[15:0]	R/W	Sets the clipping phase of the angle of THETA when phase interpolation is performed (16-bit fixed-point data from 0.0 to 1.0). Setting value: Phase [deg] ÷ 360 × 2^{16} This bit is used in SIN/COS calculation (Task 5) when phase interpolation is enabled.

18.4 Description of Operations

18.4.1 Schedule Management

Figure 18-8 shows a flowchart of motor control. The vector engine changes operation status according to the schedule setting (VExACTSCH) and mode setting (VExMODE).

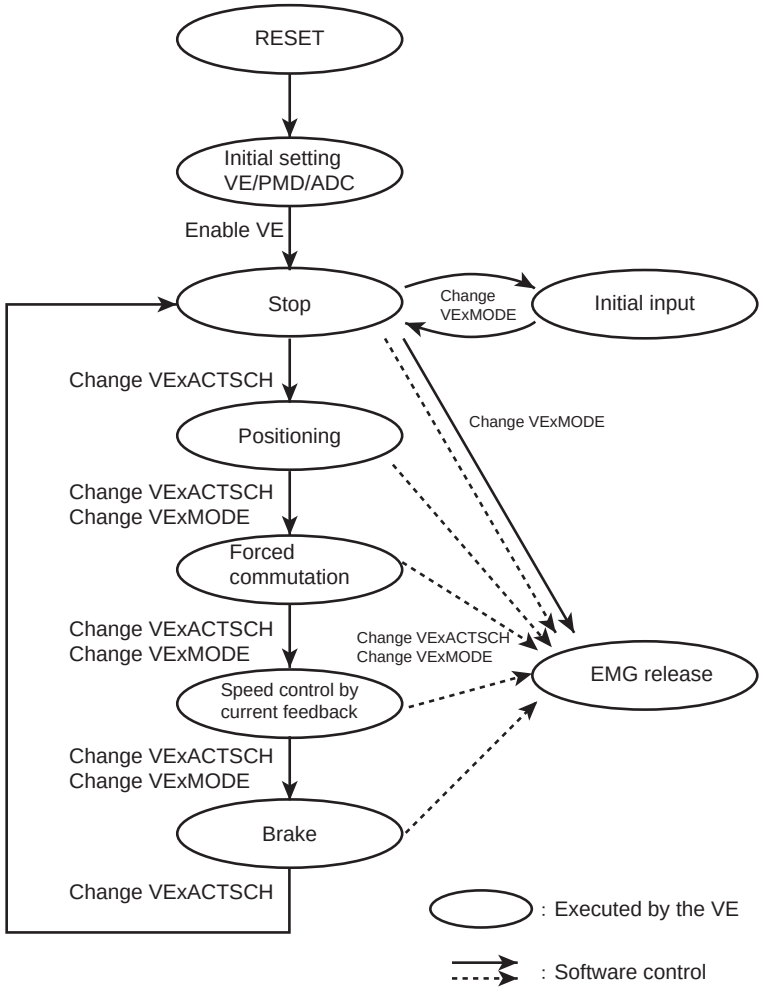


Figure 18-8 Example of Motor Control Operation Status Flowchart

RESET	: Resets the MCU.
Initial setting	: Specifies the initial setting by a user-created program.
Stop	: Stops the motor.
Initial input	: Samples and stores zero-current data when the motor is at stop.
Positioning	: Controls the motor initial position.
Forced commutation	: Rotates the motor. The motor is rotated at a specified speed for a specified period without current feedback control.
Speed control by current feedback	: Control motor rotation by current feedback.
Brake	: Deceleration control
EMG recovery	: Release the EMG protection state.

18.4.1.1 Schedule Control

An operation schedule is selected with the VExACTSCH register.

The schedule is comprised of the output schedule handling output process tasks and the input schedule handling input process tasks. Table 18-4 shows the relationship between the schedules and operation tasks.

A task operation is specified with the dedicated register according to the motor control method.

Table 18-4 Execution task in each schedule

Schedule Selection VExACTSCH <VACT[3:0]>	Output schedule execution tasks								Input schedule execution tasks				Provided by only individual execution	
	Current control	SIN/COS	Output coordinate axis transformation	Output phase transformation1	Output phase transformation2	Output control1	Output control2	Trigger generation	Input processing 1	Input processing 2	Input phase transformation	Input coordinate axis transformation	ATAN2	Square root
	Task 5	Task 6	Task 7	Task 8	Task 11	Task 0	Task 9	Task 1	Task 2	Task 10	Task 3	Task 4	Task 12	Task 13
0:Individual task execution	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)	(Note 1)
1: Schedule 1	o	o	o	o	−	o	−	o	o	−	o	o	−	−
2: Schedule 2	o	o	o	o	−	−	o	o	−	o	o	o	−	−
3: Schedule 3	o	o	o	−	o	−	o	o	−	o	o	o	−	−
4: Schedule 4	−	o	o	o	−	o	−	o	o	−	o	o	−	−
5: Schedule 5	−	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: Schedule 10	o	o	o	o	−	o	−	−	−	−	−	−	−	−
11: Schedule 11	o	o	o	o	−	−	o	−	−	−	−	−	−	−
12: Schedule 12	o	o	o	−	o	−	o	−	−	−	−	−	−	−
13: Schedule 13	o	o	o	−	o	o	−	−	−	−	−	−	−	−
14: Schedule 14	−	−	−	−	−	−	−	o	o	−	o	o	−	−
15:Schedule 15	−	−	−	−	−	−	−	o	−	o	o	o	−	−

Note 1: Only the tasks that are specified with VExTASKAPP are executed.
Note 2: o: Execution task, −: Non execution task

Table 18-5 Example of setting for typical operation flow

Setting / Motor control flow	Schedule setting VExACTSCH <VACT[3:0]>	Task selection VExTASKAPP <VTASK[3:0]>	Phase interpolation enable VExMODE <PVIEN>	Output control operation VExMODE <OCRMD[1:0]>	Zero-current detection VExMODE <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
Current feedback speed control	1	5	1	01	0
Brake	4	6	0	01	0
EMG recovery	9	0	x	11	0
Short circuit brake	4	6	x	10	0

An output schedule starts operation with the command (VExCPURUNTRG). When all output-related tasks are completed, the vector engine enters standby state and waits for a start trigger (VExTRGMODE setting).

An input schedule starts operation by a start trigger. When all input-related tasks are completed, the vector engine generates an interrupt to the CPU and enters halt state. However, if the number of repeating of the schedule (VExREPTIME) is set to "2" or more, the output schedules continue execution. An interrupt does not occur until the schedules have been executed for the specified number of times.

Note: Repeat setting is not available in Schedule 10 to Schedule 15. (The schedule ends once even if the condition is VExREPTTIME ≥ 2.)

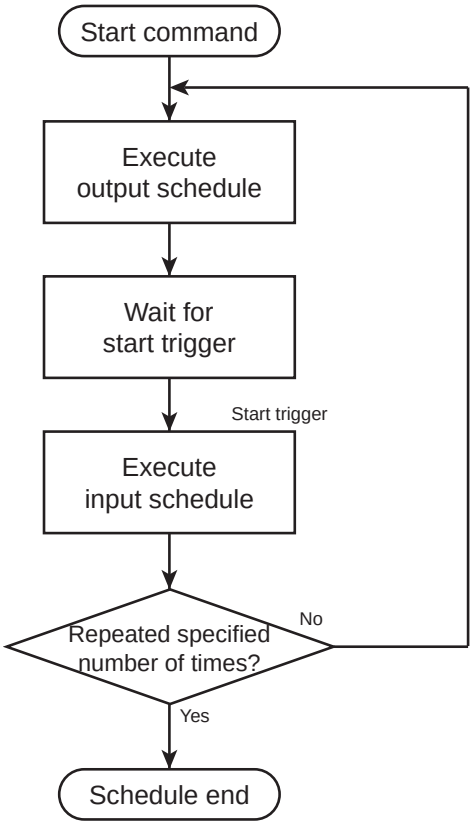


Figure 18-9 Operation schedule flowchart

18.4.1.2 Start Control

Before the schedule is started, set the vector engine to enable (<VEEN> ="1"), and then set the operation schedule selection register (VExACTSCH), task specify register (VExTASKAPP) and operation schedule repeat specify register (VExREPTIME). After that, the schedule can be executed as follows:

A schedule of the vector engine is comprised of output schedules and input schedules. Typically, the vector engine enters wait state after an output schedule is complete. At this time, if a startup trigger occurs, an input schedule is executed.

The output schedule and the input schedule are started on different conditions below:

- Start conditions for the output schedule

  1. Starting by the command. A task (VExTASKAPP) is specified with VExCPURUNTRG.

  2. Repeat setting. Starting after the input schedules have been executed for the number of times (VExREPTIME ≥ 2).

- Start conditions for the input schedule

  1. Starting an input process task that is started from wait state by a trigger (specified with VExTRGMODE). In this case, the vector engine is in wait state after the output schedule is complete.

  2. Starting by the command. A task (VExTASKAPP) is specified with VExCPURUNTRG.

Table 18-6 Schedule-related tasks

Register	Function	
VExACTSCH	Operation schedule selection	0x0: Only the task specified with VExTASKAPP is executed. 0x1: Schedule 1 is executed. 0x2: Schedule 2 is executed. 0x3: Schedule 3 is executed. 0x4: Schedule 4 is executed. 0x5: Schedule 5 is executed. 0x6: Schedule 6 is executed. 0x7: Schedule 7 is executed. 0x8: Schedule 8 is executed. 0x9: Schedule 9 is executed. 0xA: Schedule 10 is executed. 0xB: Schedule 11 is executed. 0xC: Schedule 12 is executed. 0xD: Schedule 13 is executed. 0xE:Schedule 14 is executed. 0xF: Schedule 15 is executed.
VExTASKAPP	Starting task designation	Specifies the task number that can be executed among the selected operation schedule.
VExREPTTIME	The number of repeating of the schedule	Sets the number from 1 to 15. Note: When one-time execution is specified, set "1".If "0" is set, the schedule is not executed.
VExTRGMD	Start trigger mode selection	Selects the input schedule trigger. Sets an ADC unit A interrupt or ADC unit B interrupt.

18.4.1.3 Interrupt Control

The vector engine has a vector engine interrupt (INTVCNx) that occurs at the completion of the schedule, and a task completion interrupt (INTVCTx) that occurs at the completion of specified task.

- Vector engine interrupt

  1. Specifies the schedule with the operation schedule selection register (VExACTSCH). Sets the command (VExCPURUNTRG = "1").

  2. When the schedules has been executed for the specified times (VExREPTIME), an INTVCNx interrupt occurs.

  3. If the error detection interrupt control is enabled (VExERRINTEN<VERREN> = "1"), a PWM interrupt of the PMD circuit occurs while output schedule is being executed. Then an INTVCNx interrupt occurs and an error flag (VExERRDET<VERRD>) is set to "1".

- Task completion interrupt

  1. Specifies a task triggered of a task completion interrupt (VExTASKAPP<VITASK>). Sets task completion interrupt control to enable (VExERRINTEN<INTTEN> = "1").

  2. Starts the schedule by the command (VExCPURUNTRG = "1"). Sets a INTVCTx interrupt to occur at the completion of the task specified with <VITASK>.

18.4.2 Description of Tasks

This subsection describes the outline of each task operation in the schedule.

Table 18-7 shows the task numbers that are used to specify the execution task and the start task.

Table 18-7 List of tasks

Task		Task function	Task number
Output schedule	Current control	PI control for d-axis/q-axis (PI control output limit enable) Non-interference control for d-axis and q-axis, voltage scalar limitation	5
	SIN/COS calculation	Sine/cosine calculation Phase interpolation (with clipping function)	6
	Output coordinate axis transformation	Inverse Park transformation	7
	Output phase transformation 1	Transforms from 2-phase to 3-phase [SVM]	8
	Output phase transformation 2	Transforms from 2-phase to 3-phase [Inverse Clarke transformation]	11
	Output control 1	Converts the data to PMD setting format. Switches PWM Shift 1. Limits the PWM output. Sets the dead time compensation control.	0
	Output control 2	Converts the data to PMD setting format. Switches PWM Shift 2. Limits the PWM output. Sets the dead time compensation control.	9
	Trigger generation	Generates the synchronous trigger timing.	1
Input schedule	Input process 1	Corresponds to sensor, 3-shunt and 1-shunt current detection.1-shunt is corresponded in the case of PWM shift prohibited or PWM shift 1. Captures the result of AD conversion and converts them into fixed-point format. Specifies the hysteresis width to determine the current polarity.	2
	Input process 2	Corresponds to sensor, 3-shunt and 1-shunt current detection.1-shunt is corresponded in the case of PWM shift 2. Captures the result of AD conversion and converts them into fixed-point format. Specifies the hysteresis width to determine the current polarity.	10
	Input phase transformation	Transforms from 3-phase to 2-phase.	3
	Input coordinate axis transformation	Performs Park transformation Calculates the declination angle of the current vector or the induced voltage vector on the d-q coordinate.	4
ATAN2 calculation		Calculates the arc tangent.	12
SQRTcalculation		Calculates the square root.	13

18.4.2.1 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 the d-axis and q-axis voltages respectively.

The expansion control enables the non-interference control and the voltage scalar limitation that controls p-axis and q-axis together.

1. d-axis current PI control

<Equation>

[PI control]

CIDKP = VExCIDKP × [VExCIDKG setting] : Proportional coefficient

CIDKI = VExCIDKI × [VExCIDKG setting] : Integral coefficient

Δ ID = VExIDREF − VExID : Calculates between the current reference value and the current feedback.

VDI0 = CIDKI × Δ ID + VExVDI : Calculates the integral components.

VD0 = CIDKP × Δ ID + VExVDI0 : Calculates the voltage by adding the proportional components.

[PI control output limitation]

if (VD0 > VExPIOLIM) : The upper-limit value

    VExVD = VExPIOLIM

    VExMCTLF<PIDOVF> = 1

else if (VD0 < −VExPIOLIM) : The lower-limit value

    VExVD = −VExPIOLIM

    VExMCTLF<PIDOVF> = 1

else VExVD = VD0

[Anti-windup (AWU)]

Δ VD = VExVD − VD0 :Calculates the difference between the d-axis voltage and the limit.

VExVDI = VDI0 + Δ VD × [VExMODE<AWUMD> setting] : Reflects the above difference to the integral component.

	Register name	Function	
Input	VExID	d-axis current	32-bit fixed-point data (31 fractional bits)
	VExIDREF	d-axis current reference value	16-bit fixed-point data(15 fractional bits)
	VExCIDKP	Proportional coefficient	16-bit data
	VExCIDKI	Integral coefficient	16-bit data
	VExCIDKG	PI control coefficient range setting for d-axis	000: 1/1, 001: $1/2^4$, 010: $1/2^8$, 011: $1/2^{12}$ 100: $1/2^{16}$, 101 to 111: Reserved
	VExPIOLIM	PI control output limitation	16-bit fixed-point data (15 fractional bits) Valid range: 0x0 to 0x7FFF Note: Output limitation is disabled when VExPIOLIM = 0.
	VExMODE[9:8]	Anti-windup ratio setting when output is limited.	<AWUMD> 00: Disabled, 01: 1/4, 10: 1/2, 11: 1
Output	VExVD	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	VExMCTLF[8]	d-axis output limited state	<PIDOVF> 0: No limitation, 1: Limited
Internal	VExVDI	d-axis voltage integral components are stored.	64-bit fixed-point data (63 fractional bits)

Note: The VExVDI is comprised of 64 bits. The upper is used for the VExVDIH register and the lower is used for the VExVDILH register.

2. q-axis current PI control

<Equation>

[PI control]

CIQKP = VExCIQKP × [VExCIQKG setting] : Proportional coefficient

CIQKI = VExCIQKI × [VExCIQKG setting] : Integral coefficient

Δ IQ = VExIQREF − VExIQ : Calculates between the current reference value and the current feedback.

VQI0 = CIQKI × Δ IQ + VExVQI : Integral component calculation

VQ0 = CIQKP × Δ IQ + VQI0 : Calculates the voltage by adding proportional components.

[PI control output limitation]

if (VQ0 > VExPIOLIM) : Confirm the upper-limit.

VExVQ = VExPIOLIM

VExMCTLF<PIQOVF> = 1

else if (VQ0 < −VExPIOLIM) : Confirm the lower-limit.

VExVQ = −VExPIOLIM

VExMCTLF<PIQOVF> = 1

else VExVQ = VQ0

[Anti-windup (AWU)]

Δ VQ = VExVQ − VQ0 : Calculates the difference between the q-axis voltage and the limit.

VExVQI = VQI0 + Δ VQ × [VExMODE<AWUMD> setting] : Reflects the above difference to the integral component.

	Register name	Function	
Input	VExIQ	q-axis current	32-bit fixed-point data (31 fractional bit)
	VExIQREF	q-axis current reference value	16-bit fixed-point data (15 fractional bits)
	VExCIQKP	Proportional coefficient	16-bit data
	VExCIQKI	Integral coefficient	16-bit data
	VExCIQKG	PI control coefficient range setting for q-axis	000: 1/1, 001: $1/2^4$, 010: $1/2^8$, 011: $1/2^{12}$ 100: $1/2^{16}$, 101 to 111: Reserved
	VExPIOLIM	The limit value of PI control output	16-bit fixed-point data (15 fractional bits) Valid range: 0x0 to 0x7FFF Note: Output limitation is disabled when VExPIOLIM = 0.
	VExMODE[9:8]	Anti-windup ratio setting when output is limited.	<AWUMD> 00: Disabled, 01: 1/4, 10: 1/2, 11: 1
Output	VExVQ	q-axis voltage	32-bit fixed-point data (31 fractional bits)
	VExMCTLF[9]	q-axis output limited state	<PIQOVF> 0: No limitation, 1: Limited
Internal	VExVQI	q-axis voltage integral component is stored.	64-bit fixed-point data (63 fractional bits)

Note: The VExVQI is comprised of 64 bits. The upper is used for the VExVQIHregister and the lower is used for the VExVQILH register.

3. Non-interference control

The result of PI control is corrected using the results of the interference to d-axis and q-axis based on motor voltage equation.

<Equation>

```
if (VExMODE[10] = 1 )                                          : Expansion control is enabled.
    LD = VExCLD × [VExCLG setting]                             : d-axis inductance
    LQ = VExCLQ × [VExCLG setting]                             : q-axis inductance
    PHI = VExCPHI × [VExCPHIG setting]                         : Interlinkage magnetic flux
    VExVDE = −VExOMEGA × VExIQ × LQ                            : Calculates the interference to d-axis.
    VExVQE = VExOMEGA × VExID × LD + VExOMEGA × PHI             : Calculates the interference to q-axis.
    if (VExMODE[11] = 1 )                                      : Non-interference control is enabled.
        VExVD = VExVD + VExVDE
        VExVQ = VExVQ + VExVQE
```

	Register name	Function	
Input	VExVD	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	VExVQ	q-axis voltage	
	VExID	d-axis current	
	VExIQ	q-axis current	
	VExCLD	d-axis inductance	16-bit fixed-point data (11 fractional bits)
	VExCLQ	q-axis inductance	
	VExCPHI	Interlinkage magnetic flux	
	VExCLG	Inductance range setting	000: 1/1, 001: $1/2^4$, 010: $1/2^8$, 011: $1/2^{12}$ 100: $1/2^{16}$, 101 to 111: Reserved
	VExCPHIG	Interlinkage flux range setting	
	VExOMEGA	Rotational speed	16-bit fixed-point data (15 fractional bits)
	VExMODE[11]	Non-interference control enable	<NICEN> 0: Non-interference control is disabled. 1: Non-interference control is enabled.
	VExMODE[10]	Expansion control enable	<T5ECEN> 0: Expansion control is disabled. (Non-interference control is disabled.) 1: Expansion control is enabled.
Output	VExVDE	Non-interference correction voltage for d-axis	16-bit fixed-point data (15 fractional bits)
	VExVQE	Non-interference correction voltage for q-axis	
	VExVD	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	VExVQ	q-axis voltage	

4. Voltage scalar limitation

The voltage scalar limitation controls d-axis voltage and q-axis voltage, so as to be lower the composite value (the square root of $VD^2 + VQ^2$) that is comprised of d-axis voltage and q-axis voltage than the limit value.

<Equation>

```
if (VExMODE[10] = 1 )                                                    : Expansion control is enabled.
    [VDQ calculation]
    if (VExVD² + VExVQ² > VExVSLIM² )                                    : Confirm the excess.
        if (VExFMODE[11:10] = 00)        VExVDQ = SQRT(VD² + VQ² )       : Voltage scalar limitation is disabled.
        else if (VExFMODE[11:10] =01)    VExVDQ = SQRT(VExVSLIM² − VQ²)  : Scalar limitation on the d-axis direction
        else if (VExFMODE[11:10] =10)    VExVDQ = SQRT(VExVSLIM² − VD²)  : Scalar limitation on the q-axis direction
        else if (VExFMODE[11:10] =11)    VExVDQ = SQRT(VD² + VQ² )       : dq proportional scalar limitation
        Note: SQRT is the square root calculation.
    [Calculation for the declination angle]
    X = | VExVQ |
    Y = | VExVD |
    VExDELTA = ATAN2(X, Y)
        Note: ATAN2 is arc tangent calculation.
    [Limitation calculation for each axis]
    if (VExFMODE[11:10] = 00)                                            : Voltage scalar is disabled.
        VDLIM = VExVSLIM
        VQLIM = VExVSLIM
    else if (VExFMODE[11:10] = 01)                                       : Scalar limitation on the d-axis direction
        VDLIM = VExVDQ
        VQLIM = VExVSLIM
    else if (VExFMODE[11:10] = 10)                                       : Scalar limitation on the q-axis direction
        VDLIM = VExVSLIM
        VQLIM = VExVDQ
    else if (VExFMODE[11:10] = 11)                                       : dq proportional scalar limitation
        VDLIM = VExVSLIM × SIN(VExDELTA)
        VQLIM = VExVSLIM × COS(VExDELTA)
    [Limitation process]
    if (VExVD > VDLIM)               VExVD = VDLIM                       : Process for the upper-limit of d-axis
                                     VExMCTLF[10] = 1
    else if (VExVD < −VDLIM)         VExVD = −VDLIM                      : Process for the lower-limit of d-axis
                                     VExMCTLF[10] = 1
    if (VExVQ > VQLIM)               VExVQ = VQLIM                       : Process for the upper-limit of q-axis
                                     VExMCTLF[10] = 1
```

else if (VExVQ < −VQLIM)	VExVQ = −VQLIM VExMCTLF[10] = 1	: Process for the lower-limit of q-axis

	Register name	Function	
Input	VExVD	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	VExVQ	q-axis voltage	
	VExVSLIM	The limit value of voltage scalar	16-bit fixed-point data (15 fractional bits) 0x0 to 0x7FFF Note: Limitation is disabled when VExVSLIM = 0x0.
	VExMODE[10]	Expansion control enable	<T5ECEN> 0: Expansion control is disabled. (Scalar limitation is disabled.) 1: Expansion control is enabled.
	VExFMODE[11:10]	Limitation mode setting	<VSLIMMD> 00: Scalar limitation is disabled. (Limitation in each axis is enabled.) 01: Scalar limitation is enabled. Limitation on the d-axis direction. 10: Scalar limitation is enabled. Limitation on the q-axis direction. 11: Scalar limitation is enabled. dq proportional limitation.
Output	VExVDQ	The value of voltage scalar or the value of axis direction limitation.	16-bit fixed-point data (15 fractional bits)
	VExDELTA	The declination angle	16-bit data 0x0000 to 0x4000 (0 to 90°)
	VExVD	d-axis voltage	32-bit fixed-point data (31 fractional bits)
	VExVQ	q-axis voltage	
	VExMCTLF[10]	Indicates the excess flag for voltage scalar limitation.	<VSOVF> 0:non excess 1:excess

18.4.2.2 SIN/COS calculation (Task 6)

The SIN/COS calculation task executes phase interpolation calculation and SIN/COS calculation.

Phase interpolation calculates the rotation speed by integrating with the PWM period. It is executed only when phase interpolation is enabled (VExMODE<PVIEN> = "1").

1. Phase interpolation

<Equation>

```
THETA0 = VExOMEGA × VExTPWM + VExTHETA                                  :Calculates the value of phase interpolation.
THETA0 = THETA0 & 0x0000FFFF
if (VExMODE[7] = 1)                                                     : Enables the clipping
    if ( VExOMEGA ≥ 0)                                                  : On positive rotation.
        if ( VExTHETA ≤ VExTHTCLP ≤ THETA0 )        THETA0 = THTCLP
        else if ( THETA0 ≤ VExTHETA ≤ VExTHTCLP )   THETA0 = THTCLP
        else if ( VExTHTCLP ≤ THETA0 ≤ VExTHETA )   THETA0 = THTCLP
    else if ( VExOMEGA < 0)                                             :On inverse rotation.
        if ( THETA0 ≤ VExTHTCLP ≤ VExTHETA )        THETA0 = THTCLP
        else if ( VExTHTCLP ≤ VExTHETA ≤ THETA0 )   THETA0 = THTCLP
        else if ( VExTHETA ≤ THETA0 ≤ VExTHTCLP )   THETA0 = THTCLP
if ( VExMODE[0] = 1 ) VExTHETA = THETA0                                 : Updates the THETA value when phase interpolation is enabled.
```

	Register name	Function	
Input	VExTHETA	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
	VExOMEGA	Rotational speed	16-bit fixed-point data (−1.0 to 1.0, 15 fractional bits)
	VExTPWM	PWM cycle rate	16-bit data
	VExTHTCLP	The value of clipping	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
	VExMODE[0]	Phase interpolation enable	<PVIEN> 0: Phase interpolation is disabled. 1: Phase interpolation is enabled.
	VExMODE[7]	Phase clipping control	<CLPEN> 0: Clipping is disabled. 1: Clipping is enabled.
Output	VExTHETA	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)

2. SIN/COS calculation

<Equation>

```
VExSINM = VExSIN                          : Stores the previous value (for input process)
VExCOSM = VExCOS
VExSIN = SIN ( VExTHETA )                 : Calculates the SIN and COS values.
VExCOS = SIN ( VExTHETA + 1/4 )
if ( VExFMODE[9] = 1)                     : Confirms that the previous value is not maintained.
    VExSINM = VExSIN
    VExCOSM = VExCOS
```

Note: SIN: Sine calculation

	Register name	Function	
Input	VExTHETA	Phase θ	16-bit fixed-point data (0.0 to 1.0, 16 fractional bits)
	VExFMODE[9]	Selects to store the previous value of SIN and COS	<MREGDIS> 0: Previous value is maintained. 1: Previous value is not maintained.
Output	VExSIN	The sine value of θ	16-bit fixed-point data (−1.0 to 1.0, 15 fractional bits)
	VExCOS	The cosine value of θ	
	VExSINM	Previous sine value	
	VExCOSM	Previous cosine value	

18.4.2.3 Output Voltage Transformation (Coordinate Axis Transformation/Phase Transformation)

Output voltage transformation is performed in two steps: coordinate axis transformation and phase transformation. There are two types of phase transformation: Space vector transformation 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 based on d-axis voltage, q-axis voltage, sinθ and cosθ.

   <Equation>

   VExTMPREG3 = VExCOS × VExVD − VExSIN × VExVQ : Calculates Vα

   VExTMPREG4 = VExSIN × VExVD + VExCOS × VExVQ : Calculates Vβ

	Register name	Function	
Input	VExVD	d-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExVQ	q-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExSIN	Sine value of THETA	16-bit fixed-point data (−1.0 to 1.0, 15 fractional bits)
	VExCOS	Cosine value of THETA	16-bit fixed-point data (−1.0 to 1.0, 15 fractional bits)
Output	VExTMPREG3	α-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)

2. Output phase transformation 1 (Space vector transformation) (Task 8)

   Output phase transformation 1 determines a sector using α-axis voltage and β-axis voltage. This task calculates the duties of a-phase voltage, b-phase voltage, and c-phase voltage based on space vector transformation in each sector. This task can be selected either from 2-phase or 3-phase modulation as a modulation type.

a. Sector determination

<Equation>

```
VExSECTORM = VExSECTOR                          : Stores the previous sector.
Vα = VExTMPREG3
Vβ = VExTMPREG4
if ( Vα ≥ 0 & Vβ ≥ 0 )
    if ( |Vα| ≥ |Vβ| ÷ sqrt(3) )
        if ( |Vα| ÷ sqrt(3) ≥ |Vβ| )    VExSECTOR = 0
        else                            VExSECTOR = 1
    else                                VExSECTOR = 2
else if ( Vα < 0 & Vβ ≥ 0 )
    if ( |Vα| < |Vβ| ÷ sqrt(3) )        VExSECTOR = 3
    if ( |Vα| ÷ sqrt(3) < |Vβ| )        VExSECTOR = 4
    else                                VExSECTOR = 5
else if ( Vα < 0 & Vβ < 0 )
    if ( |Vα| ≥ |Vβ| ÷ sqrt(3) )
        if ( |Vα| ÷ sqrt(3) ≥ |Vβ| )    VExSECTOR = 6
        else                            VExSECTOR = 7
    else                                VExSECTOR = 8
else if ( Vα ≥ 0 & Vβ < 0 )
    if ( |Vα| < |Vβ| ÷ sqrt(3) )        VExSECTOR = 9
    else if ( |Vα| ÷ sqrt(3) < |Vβ| )   VExSECTOR = 10
    else                                VExSECTOR = 11
if ( VExFMODE[9] = 1) VExSECTORM = VExSECTOR    : Check whether the previous value is invalid.
```

	Register name	Function	
Input	VExTMPREG3	α-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExFMODE[9]	Selects to store the previous value.	<MREGDIS> 0: The previous sector value is maintained. 1: The previous sector value is not maintained.
Output	VExSECTOR	Sector	4-bit data
	VExSECTORM	Previous sector	4-bit data

b. Space vector transformation (This example describes only the case where 3-phase modulation is used and <SECTOR[3:0]> = "0" and "1".)

<Equation>

```
if (SECTOR = 0,1)
    t1 = sqrt(3) ÷ VExVDC × (sqrt(3) ÷ 2 × Vα − 1 ÷ 2 × Vβ)      : Calculates the t1 period.
    t2 = sqrt(3) ÷ VExVDC × Vβ)                                  : Calculates the t2 period.
    t3 = 1 − t1 − t2                                             : Calculates the zero-vector period.
    if ( VExFMODE[0] = 0 )                                       : 3-phase modulation
        DUTYA = t1 + t2 + t3 ÷ 2
        DUTYB = t2 + t3 ÷ 2
        DUTYC = t3 ÷ 2
    else                                                         : 2-phase modulation
        DUTYA = t1 + t2
        DUTYB = t2
        DUTYC = 0

VExTMPREG0 = DUTYA
VExTMPREG1 = DUTYB
VExTMPREG2 = DUTYC
VExTMPREG5 = t3
```

	Register name	Function	
Input	VExTMPREG3	α-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis voltage	
	VExVDC	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExSECTOR	Sector	4-bit data
	VExFMODE[0]	Modulation mode	<C2PEN> 0: 3-phase modulation 1: 2-phase modulation
Output	VExTMPREG0	a-phase voltage Duty	32-bit fixed-point data (0..0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase voltage Duty	
	VExTMPREG2	c-phase voltage Duty	
	VExTMPREG5	Zero-vector duty	

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 calculates the duties of a-phase, b-phase and c-phase voltages based on inverse Clarke transformation . This task supports only 3-phase modulation as a modulation type.

In addition, if VExFMODE<PHCVDIS> is set to "1", this task can calculate the duty of 2-phase voltage.

a. Sector determination

<Equation>

```
VExSECTORM = VExSECTOR                          : Stores the previous sector.
Vα = VExTMPREG3
Vβ = VExTMPREG4
if ( Vα ≥ 0 & Vβ ≥ 0 )
    if ( Vα| ≥ |Vβ| ÷ sqrt(3))
        if ( |Vα| ÷ sqrt(3) ≥ |Vβ| )    VExSECTOR = 0
        else                            VExSECTOR = 1
    else                                VExSECTOR = 2
else if ( Vα < 0 & Vβ ≥ 0 )
    if ( |Vα| < |Vβ| ÷ sqrt(3) )        VExSECTOR = 3
    if ( |Vα| ÷ sqr(3) < |Vβ| )         VExSECTOR = 4
    else                                VExSECTOR = 5
else if ( Vα < 0 & Vβ < 0 )
    if ( |Vα| ≥ |Vβ| ÷ sqrt(3) )
        if ( |Vα| ÷ sqrt(3) ≥ |Vβ| )    VExSECTOR = 6
        else                            VExSECTOR = 7
    else                                VExSECTOR = 8
else if ( Vα ≥ 0 & Vβ < 0 )
    if ( |Vα| < |Vβ| ÷ sqrt(3) )        VExSECTOR = 9
    else if ( |Vα| ÷ sqrt(3) < |Vβ| )   VExSECTOR = 10
    else                                VExSECTOR = 11
if ( VExFMODE[9] = 1) VExSECTORM = VExSECTOR    : Confirm that the previous value is disabled.
```

	Register name	Function	
Input	VExTMPREG3	α-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExFMODE[9]	Selects to store the previous value of SECTOR	<MREGDIS> 0: Previous value is maintained. 1: Previous value is not maintained.
Output	VExSECTOR	Sector	4-bit data
	VExSECTORM	Previous sector	4-bit data

b. Inverse Clarke transformation

<Equation>

```
if ( VExFMODE[12] = 0 )                                                    : 3-phase conversion is enabled.
    VExTMPREG0 = 1 ÷ VExVDC × Vα + 1/2                                     : Va Duty
    VExTMPREG1 = 1 ÷ VExVDC × ( −1 ÷ 2 × Vα + sqrt(3) ÷ 2 × Vβ) + 1/2      : Vb Duty
    VExTMPREG2 = 1 ÷ VExVDC × ( −1 ÷ 2 × Vα − sqrt(3) ÷ 2 × Vβ) + 1/2      : Vc Duty
else                                                                       : Phase conversion is disabled.
    VExTMPREG0 = 1 ÷ VExVDC × Vα + 1/2                                     : Va Duty
    VExTMPREG1 = 1 ÷ VExVDC × Vβ + 1/2                                     : Vb Duty
```

	Register name	Function	
Input	VExTMPREG3	α-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis voltage	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExVDC	Supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExFMODE[12]	Phase conversion setting	<PHCVDIS> 0: Phase conversion is enabled. 1: Phase conversion is disabled.
Output	VExTMPREG0	a-phase voltage Duty	32-bit fixed-point data (0..0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase voltage Duty	32-bit fixed-point data (0..0 to 1.0, 31 fractional bits)
	VExTMPREG2	c-phase voltage Duty	32-bit fixed-point data (0..0 to 1.0, 31 fractional bits)

18.4.2.4 Output Control

Output control unit converts a 3-phase voltage duty into the PMD setting format. The conversion result is set to VExCMPU, VExCMPV, and VExCMPW. According to the output control setting, VExOUTCR should be set. Dead time compensation control and PWM output limitation can also be executed.

There are two types of output control task: output control 1 and output control 2. 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 is enabled, if the rotational speed (VExOMEGA) is lower than the reference of PWM shift switch (VExFPWMCHG), PWM outputs changes to PWM Shift 1.

   Note: PWM Shift can only be selected in the 1-shunt current detection mode.

   - Output conversion

     <Equation>

```
VExMCTLF[1] = VExMCTLF[0]                          :The previous value flag is updated.
VExMCTLF[0] = 0                                    :Current flag is cleared.
if ((FMODE[3] = 1) & (FMODE[0] = 1) & (FMODE[1] = 1))   :Shift 1 is enabled in 1-shunt current and 2-phase modulation modes.
    if (|VExOMEGA| < VExFPWMCHG)  VExMCTLF[0] = 1   :Sets the low speed flag for low-speed determination.
DUTYA = VExTMPREG0
DUTYB = VExTMPREG1
DUTYC = VExTMPREG2
if (VExMCTLF[0] = 1)                               :PWM Shift 1 at low speed
    if (VExSECTOR = 0,3,4,7,8,11)                  :Sector determination
        DUTYA = DUTYA + VExTMPREG5                 :Zero-vector V7 transformation
        DUTYB = DUTYB + VExTMPREG5
        DUTYC = DUTYC + VExTMPREG5
PWMA = VExTMPREG0 × VExMDPRD                       :Transforms the PMD setting value.
PWMB = VExTMPREG1 × VExMDPRD
PWMC = VExTMPREG2 × VExMDPRD
```

	Register name	Function	
Input	VExTMPREG0	a-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase voltage	
	VExTMPREG2	c-phase voltage	
	VExMDPRD	PWM cycle setting	16-bit data (PWMcycle setting value in thePMD.)
	VExSECTOR	Sector	4-bit data
	VExOMEGA	Rotational speed	16-bit fixed-point data (−1.0 to 1.0, 15 fractional bits)
	VExFPWMCHG	PWMshift switch reference speed	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExFMODE[0]	Modulation mode	<C2PEN> 0: 3-phase modulation, 1: 2-phase modulation
	VExFMODE[1]	PWM shift enable	<SPWMEN> 0: Shift is disabled. 1: Shift is enabled.
	VExFMODE[3:2]	Current detection mode	<IDMODE[1:0]> 00: 3-shunt. 01: 2 sensors, 1x: 1-shunt
Output	VExMCTLF[1:0]	Low-speed flag	<LAVFM>,<LAVF>

- PWM output limitation

<Equation>

```
if (VExPWMMAX = 0)                    MAX = VExMDPRD
else                                  MAX = VExPWMMAX
if ((PWMA > MAX) & (VExMDPRD > MAX))                  :Checks the PWM upper-limit for U-phase.
    if ((VExMODE[13] = 0) | (PWMA < VExMDPRD))        :Confirms the duty of 100% of the output limitation.
        PWMA = MAX
        VExMCTLF[11] = 1
MIN = VExPWMMIN
if ((PWMA < MIN) & (MIN > 0))                         :Checks the PWM lower-limit for U-phase.
    if ((VExMODE[12] = 0) | (PWMA > 0))               :Confirms the duty of 0% of the output limitation.
        PWMA = MIN
        VExMCTLF[11] = 1
(Calculates the other 2-phase in the same manner.)
```

	Register name	Function	
Input	VExMDPRD	PWM cycle setting	16-bit data (PWMcycle setting value in thePMD.)
	VExPWMMAX	Sets the upper-limit value of the PWM.	16-bit data (The setting value is from 0 to VExMDPRD.)
	VExPWMMIN	Sets the lower-limit value of the PWM.	
	VExMODE[12]	Sets the output duty of 0% when the PWM is limited.	<PWMBLEN> 0: Disabled, 1: Enabled
	VExMODE[13]	Sets the output duty of 100% when the PWM is limited.	<PWMFLEN> 0: Disabled, 1: Enabled
Output	VExMCTLF[11]	Indicates the excess flag of the PWM output limitation.	<PWMOVF>

- Dead time compensation

<Equation>

```
if (0 < PWMA < VExMDPRD)                DT = VExDTC
else                                    DT = 0
if (VExDTCS<IASTS> = 01)                                    :On Positive current
    if (VExMODE[14] = 1)                                    Sets the dead time correction of the PMD.
        if (PWMA > (VExMDPRD − 2× DT))  PWMA = (VExMDPRD + PWMA) ÷2
    else                                PWMA = PWMA + DT
    if ((VExMODE[13] = 1) & (PWMMAX < VExMDPRD))            :Confirms the duty of 100% of the output limitation.
        if (PWMA > (VExMDPRD −1))       PWMA = VExMDPRD −1  :Output limitation after correction.
    else
        if (PWMA > VExMDPRD)            PWMA = VExMDPRD     :Output limitation after correction.
else if (VExDTCS<IASTS> = 11)                               :On negative current
    if (VExMODE[14] = 1)                                    :Sets the dead time correction of the PMD.
        if (PWMA < (2 × DT))            PWMA = PWMA ÷2
    else                                PWMA = PWMA − DT
    if ((VExMODE[12] = 1) & (PWMMIN > 0))                   :Confirms the duty of 0% of the output limitation.
        if (PWMA < 1)                   PWMA = 1            :Output limitation after correction.
    else
        if (PWMA < 0)                   PWMA = 0            :Output limitation after correction.
( Calculates the other 2-phase in the same manner.)
```

	Register name	Function	
Input	VExMDPRD	PWM cycle setting	16-bit data (The value of the PMD PWMcycle setting.)
	VExDTC	The amount of dead time compensation	16-bit data (The setting value is from 0 to VExMDPRD.)
	VExMODE[12]	Output enable of 0% when the limitation is enabled.	<PWMBLEN>
	VExMODE[13]	Output enable of 100% when the limitation is enabled.	<PWMFLEN>
	VExMODE[14]	Dead time correction control of the PMD circuit.	<PMDDTCEN> 0: PMD dead time correction is disabled. 1: PMD dead time correction is enabled.
	VExDCTS	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> x0: Undefined, 01: Positive current, 11: Negative current

- Output control /PWM Shift 1 transformation

<Equation>

```
OUTCR = 0x1FF                                                    : All phases center ON
if (VExMCTLF[0] = 1)                                             : PWM Shift 1 at low speed.
    if (VExSECTOR = 0,1,2,11)        PWMB = MDPRD − PWMB         : V-phase center OFF
                                     OUTCR = 0x1F3
    else if (VExSECTOR = 3,4,5,6)    PWMC = MDPRD − PWMC         : W-phase center OFF
                                     OUTCR = 0x1CF
    else if (VExSECTOR = 7,8,9,10)   PWMA = MDPRD − PWMA         : U-phase center OFF
                                     OUTCR = 0x1FC
if (VExMODE[3:2] = 00,11)            OUTCR = 0x000               : Output OFF
else if (VExMODE[3:2] = 10)          OUTCR = 0x015               : Short circuit brake
VExCMPU = PWMA
VExCMPV = PWMB
VExCMPW = PWMC
VExOUTCR = OUTCR
```

	Register name	Function	
Input	VExSECTOR	Sector	4-bit data
	VExMODE[3:2]	Output control operation	<OCRMD[1:0]>
	VExMINPLS	Difference of small pulse width	16-bit data
Output	VExCMPU	U-phase PWM setting of the PMD	16-bit data (The value from 0 to MDPRD.)
	VExCMPV	V-phase PWM setting of the PMD	
	VExCMPW	W-phase PWM setting of the PMD	
	VExOUTCR	PMD output control setting	9-bit setting
	VExEMGRS	PMD EMG protection release	1-bit setting
	VExMCTLF	Small pulse flag	<PLSLF>

2. Output control 2 (Task 9)

Output control 2 task supports two mode outputs: normal PWM outputs and PWM outputs in Shift 2 mode.

When the PWM output in PWM Shift 2 mode is set, enable PWM shift (VExFMODE<SPWMEN> = "1") and select the value other than "00" for PWM shift mode selection (VExFMODE<SPWMMD>).

Note: PWM shift can be selected only in 1-shunt current detection mode.

- Output conversion

<Equation>

```
VExMCTLF[1] = VExMCTLF[0]                 : The previous value flag is updated.
VExMCTLF[0] = 0                           : Current flag is cleared.
PWMA = VExTMPREG0 × VExMDPRD              : The setting value of the PMDis converted.
PWMB = VExTMPREG1 × VExMDPRD
PWMC = VExTMPREG2 × VExMDPRD
```

	Register name	Function	
Input	VExTMPREG0	a-phase voltage	32-bit fixed-point data (0.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase voltage	
	VExTMPREG2	c-phase voltage	
	VExMDPRD	PWM cycle setting	16-bit data (The value of PWMcycle setting of the PMD)
Output	VExMCTLF[1:0]	Low-speed flag	<LAVFM>,<LAVF>

- PWM output limitation

<Equation>

```
if (VExPWMMAX = 0)                        MAX = VExMDPRD
else                                      MAX = VExPWMMAX
if ((PWMA > MAX) & (VExMDPRD > MAX))      :Checks the PWM upper-limit for U-phase.
    if ((VExMODE[13] = 1) | (PWMA < VExMDPRD))   :Confirms the duty of 100% of the output limitation.
        PWMA = MAX
        VExMCTLF[11] = 1
MIN = VExPWMMIN
if ((PWMA < MIN) & (MIN > 0))             :Checks the PWM lower-limit for U-phase.
    if ((VExMODE[12] = 1) | (PWMA > 0))   :Confirms the duty of 0% of the output limitation.
        PWMA = MIN
        VExMCTLF[11] = 1
( Calculates the other 2-phase in the same manner.)
```

	Register name	Function	
Input	VExMDPRD	PWM cycle setting	16-bit data (The value of the PMD PWM cycle setting.)
	VExPWMMAX	Sets the PWM upper-limit.	16-bit data (The value is from 0 to VExMDPRD.)
	VExPWMMIN	Sets the PWM lower-limit.	
	VExMODE	Output control operation	<OCRMD[1:0]>
	VExFMODE[12]	Output enable of 0% when the limitation is enabled.	<PWMBLEN> 0: Disabled, 1: Enabled
	VExFMODE[13]	Output enable of 100% when the limitation is enabled.	<PWMFLEN> 0: Disabled, 1: Enabled
Output	VExMCTLF[11]	PWM output limitation excess flag	<PWMOVF>

- Dead time compensation

<Equation>

```
if (0 < PWMA < VExMDPRD)              DT = VExDTC
else                                  DT = 0
if (VExDTCS<IASTS> = 01)                                  :On positive current
    if (VExMODE[14] = 1)                                  :Sets the dead time correction of the PMD.
        if (PWMA > (VExMDPRD − 2× DT))  PWMA =(VExMDPRD + PWMA)÷2
    else                              PWMA = PWMA + DT
    if ((VExMODE[13] =1) & (PWMMAX < VExMDPRD))           :Confirms the duty of 100% of the output
                                                           limitation.
        if (PWMA > (VExMDPRD −1))     PWMA = VExMDPRD −1  :Output limitation after correction.
    else
        if (PWMA > VExMDPRD)          PWMA = VExMDPRD     :Output limitation after correction.
else if (VExDTCS<IASTS> =11)                              :On negative current
    if (VExMODE[14] =1)                                   :Sets the dead time correction of the PMD.
        if (PWMA < (2× DT))           PWMA = PWMA ÷ 2
    else                              PWMA = PWMA − DT
    if ((VExMODE[12] =1) & (PWMMIN > 0))                  :Confirms the duty of 0% of the output lim-
                                                           itation.
        if (PWMA < 1)                 PWMA = 1            :Output limitation after correction.
    else
        if (PWMA < 0)                 PWMA = 0            :Output limitation after correction.
```

(Calculates the other 2-phase in the same manner.)

	Register name	Function	
Input	VExMDPRD	PWM cycle setting	16-bit data (PWM cycle setting value in the PMD.)
	VExDTC	The amount of dead time compensation	16-bit data (Setting value is from 0 to VExMDPRD.)
	VExMODE[12]	Output enable of 0% when the limitation is enabled.	<PWMBLEN]>
	VExMODE[13]	Output enable of 100% when the limitation is enabled.	<PWMFLEN]>
	VExMODE[14]	Dead time correction control of the PMD circuit.	<PMDDTCEN> 0: PMD dead time correction is disabled. 1: PMD dead time correction is enabled.
	VExDCTS	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> x0: Undefined, 01: Positive current, 11: Negative current

- Output control /PWM Shift 2 transformation

<Equation>

```
OUTCR = 0x1FF                                                    : All phases center ON
if ((VExFMODE[3] = 1) & (VExFMODE[1] = 1))                       : PWM Shift 2 is enabled in 1-shunt current mode.
    if (VExFMODE[15:14] = 01)                                    : Shift 2, U-phase reference
        if (PWMB > VExMDPRD ÷2)   PWMB = VExMDPRD − PWMB         : V-phase PWM center OFF
                                  OUTCR = OUTCR & 0x1F3
        if (PWMC > VExMDPRD ÷2)   PWMC = VExMDPRD − PWMC         : W-phase PWM center OFF
                                  OUTCR = OUTCR & 0x1CF
    else if (VExFMODE[15:14] = 10)                               : Shift 2, V-phase reference
        if (PWMA > VExMDPRD ÷2)   PWMA = VExMDPRD − PWMA         : U-phase PWM center OFF
                                  OUTCR = OUTCR & 0x1FC
        if (PWMC > VExMDPRD ÷2)   PWMC = VExMDPRD − PWMC         : W-phase PWM center OFF
                                  OUTCR = OUTCR & 0x1CF
    else if (VExFMODE[15:14] = 11)                               : Shift 2, W-phase reference
        if (PWMA > VExMDPRD ÷2)   PWMA = VExMDPRD − PWMA         : U-phase PWM center OFF
                                  OUTCR = OUTCR & 0x1FC
        if (PWMB > VExMDPRD ÷2)   PWMB = VExMDPRD − PWMB         : V-phase PWM center OFF
                                  OUTCR = OUTCR & 0x1F3
if (VExMODE[3:2] = 00,11)         OUTCR = 0x000                  : Output OFF
else if (VExMODE[3:2] = 10)       OUTCR = 0x015                  : Short circuit brake
VExCMPU = PWMA
VExCMPV = PWMB
VExCMPW = PWMC
VExOUTCR = OUTCR
```

	Register name	Function	
Input	VExMDPRD	PWM cycle setting	16-bit data (PWM cycle setting value in the PMD.)
	VExFPWMCHG	PWM level shift	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExMODE[3:2]	Output control operation	<OCRMD>
	VExMINPLS	Small pulse width	16-bit data
	VExFMODE[1]	PWM shift enable	<SPWMEN>
	VExFMODE[3:2]	Current detection mode	<IDMODE>
	VExFMODE[15:14]	PWMshift mode	<SPWMMD>
Output	VExCMPU	U-phase PWM setting of the PMD	16-bit data (The value from 0 to MDPRD.)
	VExCMPV	V-phase PWM setting of the PMD	
	VExCMPW	W-phase PWM setting of the PMD	
	VExOUTCR	PMDOutput control setting	9-bit setting
	VExEMGRS	PMD EMG protection release	1-bit setting
	VExMCTLF	Small pulse flag	<PLSLF>

18.4.2.5 Trigger Generation (Task 1)

The trigger generation unit calculates the trigger timing from the PWM setting values (VExCMPU, VExCMPV, and VExCMPW) based on the current detection method, and sets it to the VExTRGCMP0 and VExTRGCMP1 registers.

Note: VExTRGCMP0 and VExTRGCMP1 are updated only in 1-shunt current detection.

Note: VExTRGCMP0, and VExTRGCMP1 are not updated when PWM Shift 2 mode is selected.

	Register name	Function	
Input	VExCMPU	PMD U-phase PWM setting	16-bit data (the value from 0 to MDPRD)
	VExCMPV	PMD V-phase PWM setting	16-bit data (the value from 0 to MDPRD)
	VExCMPW	PMD W-phase PWM setting	16-bit data (the value from 0 to MDPRD)
	VExMDPRD	PWMcycle setting	16-bit data (PMD PWM cycle setting value)
	VExTADC	ADconversion time	16-bit data (the value from 0 to MDPRD)
	VExTRGCRC	Trigger correction value	16-bit data (the value from 0 to MDPRD)
	VExSECTOR	Sector	4-bit data
	VExMODE[0]	0-current detection	<ZIEN>
	VExMODE[3:2]	Output control operation	<OCRMD>
	VExFMODE[0]	Modulation mode	<C2PEN> 0: 3-modulation, 1: 2-phase modulation
	VExFMODE[1]	PWM shift enable	<SPWMEN>
	VExFMODE[3:2]	Current detection mode	<IDMODE>
	VExFMODE[8]	Trigger correction enable	<CRCEN>
	VExFMODE[15:14]	PWM shift mode	<SPWMMD>
	VExMCTLF[0]	Low-speed flag	<LAVF>
Output	VExTRGCMP0	PMD trigger 0 timing setting	16-bit data (the value from 0 to MDPRD)
	VExTRGCMP1	PMD trigger 1 timing setting	16-bit data (the value from 0 to MDPRD)
	VExTRGSEL	PMD trigger selection	3-bit data

18.4.2.6 Input Process

In the input process, the vector engine reads conversion results and phase information from the AD converter. Depending on the current detection method and PWM Shift mode setting, the vector engine converts the phase current data and the voltage data into the fixed-point format and stores them in predefined registers. In zero-current detection mode, current detection results are stored in the zero-current register.

The vector engine specifies the hysteresis width to determine the current polarity for dead time compensation control.

There are two types of input process: Input process 1 task and Input process 2 task. Each task supports different current detection method.

1. Input process 1 (Task 2)

   Input process 1 task supports 3-shunt current detection (only 2-phase current detection (Note 1)) and 1-shunt current detection. However, 1-shunt current detection is not available in PWM Shift 2 mode (Note 2).

   Note 1: Current detection result is used only for two phases. Remained one phase is calculated.

   Note 2: PWM Shift can only be selected in 1-shunt current detection mode.

   - Input conversion

     <Equation>

```
[VDC fixed-point transformation/store]
if (VExMODE[4] = 0                          VExVDC = [DC voltage] >>1
else                                        VExVDCL = [DC voltage] >>1
[Current 1 read]
if (VExFMODE[3:2] = 10,11                                              : 1-shunt
    if (VExMCTLF[1] = 0                                                : Normal PWM
        if (VExSECTORM = 4,5,6,7)           VExIAADC = [Current 1]
        else if (VExSECTORM = 8,9,10,11)    VExIBADC = [Current 1]
        else f (VExSECTORM = 0,1,2,3)       VExICADC = [Current 1]
    else if (VExMCTLF[1] = 1                                           : PWM Shift 1
        if (VExSECTORM = 1,2,7,8)           VExIAADC = [Current 1]
        else if (VExSECTORM = 0,5,6,11)     VExIBADC = [Current 1]
        else f (VExSECTORM = 3,4,9,10)      VExICADC = [Current 1]
else if (VExFMODE[3:2] = 00,01                                         : 3-shunt, 2 sensors
    if ([Current 1 phase information] = 1)  VExIAADC = [Current 1]
    else if ([Current 1 phase information] = 2)
                                            VExIBADC = [Current 1]
    else if ([Current 1 phase information] = 3)
                                            VExICADC = [Current 1]
[Current 2 read]
if (VExFMODE[3:2] = 10,11                                              : 1-shunt
    if (VExMCTLF[1] = 0                                                : Normal PWM
        if (VExSECTORM = 0,1,10,11)         VExIAADC = [Current 2]
        else if (VExSECTORM = 2,3,4,5)      VExIBADC = [Current 2]
        else f (VExSECTORM = 6,7,8,9)       VExICADC = [Current 2]
    else if (VExMCTLF[1] = 1                                           : PWM Shift 1
        if (VExSECTORM = 3,4,9,10)          VExIAADC = [Current 2]
        else if (VExSECTORM = 1,2,7,8)      VExIBADC = [Current 2]
```

```
        else f (VExSECTORM = 0,5,6,11)      VExICADC = [Current 2]
else if (VExFMODE[3:2] = 00,01                                          : 3-shunt, 2 sensors
    N = [Current 2 phase information]                                   : Detected by 1 AD converter se-
    X = [Current 2]                                                       quentially.
    if (VExFMODE[13] = 1                    N = [Current 4 phase information]  : Current 4 is used at synchro-
                                            X = [Current 4]                      nous sampling.
        if ( N = 1)                         VExIAADC = X
        else if ( N = 2)                    VExIBADC = X
        else if ( N = 3)                    VExICADC = X
[Current 3 read]
if (VExFMODE[3] ≠ 1)                                                    : Except 1-shunt
    if ( [Current 3 phase information] = 1) VExIAADC = [Current 3]
    else if ( [Current 3 phase information] = 2)  VExIBADC = [Current 3]
    else if ( [Current 3 phase information] = 3)  VExICADC = [Current 3]
[Current fixed-point transformation]
IA = VExIAO − VExIAADC
IB = VExIBO − VExIBADC
IC = VExICO − VExICADC
if (VExFMODE[3:2] = 10,11                                               : 1-shunt
    if (VExMCTLF[1] = 0                                                 : Normal PWM
        if (VExSECTORM = 0,1,10,11)     IA = −IA
        else if (VExSECTORM = 6,7,8,9)  IB = −IB
        else f (VExSECTORM = 2,3,4,5)   IC = −IC
    else if (VExMCTLF[1] = 1                                            : PWM Shift 1
        if (VExSECTORM = 1,2,5,6,9,10)  IA = −IA
                                        IB = −IB
                                        IC = −IC
[Current 3 calculation]
N = 6 − N − [Current 1 phase information]                               : Calculates the phase number
                                                                          of Current 3.
if ( N = 1)                             IA = − IB − IC
else if ( N = 2)                        IB = − IC − IA
else if ( N = 3)                        IC = − IA − IB
[Current storage]
VExTMPREG0 = IA
VExTMPREG1 = IB
VExTMPREG2 = IC
```

	Register name	Function	
Input	(DC voltage)	Input of the ADC conversion result	16-bit data (The conversion result is stored in the upper 12 bits.)
	(Current 1)		
	(Current 2)		
	(Current 3)		
	(Current 4)		
	VExSECTORM	Sector information	4-bit data
	VExMODE[1]	Zero-current detection	<ZIEN>
	VExMODE[4]	VDCstore register	<VDCSEL>
	VExFMODE[3:2]	Current detection mode	<IDMODE>
	VExFMODE[13]	Synchronous sampling control	<SADCEN> 0: Synchronous sampling is disabled., 1: Synchronous sampling is enabled.
	VExMCTLF[0]	Low-speed flag	<LAVFM>

	Register name	Function	
Output	VExVDC	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExVDCL	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExIAADC	The result of current conversion of a-phase	16-bit data (The result is stored in the upper 12 bits.)
	VExIBADC	The result of current conversion of b-phase	
	VExICADC	The result of current conversion of c-phase	
	VExTMPREG0	a-phase current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase current	
	VExTMPREG2	c-phase current	
Internal	VExIAO	The result of zero-current conversion of a-phase	16-bit data (The result is stored in the upper 12 bits.)
	VExIBO	The result of zero-current conversion of b-phase	
	VExICO	The result of zero-current conversion of c-phase	

- Current polarity determination

<Equation>

```
if ( VExMODE[15] = 1)                                   : Current polarity determination is enabled.
    IA = VExTMPREG0
    if ( <IASTS> = xx0)                                 : The polarity is undefined.
        if ( IA ≥ |HYS| )           <IASTS> = 001       : Positive polarity is determined.
        else if ( IA ≤ −|HYS| )     <IASTS> = 111       : Negative polarity is determined.
    else if ( <IASTS> = 001)                            : The previous polarity was positive.
        if ( IA ≤ −|HYS| )          <IASTS> = 111       : Polarity is changed to negative.
        else if ( IA < −HYS )       <IASTS> = 011       : Polarity is changed to negative.(Hysteresis area)
    else if ( <IASTS> = 101)
        if ( IA ≥ |HYS| )           <IASTS> = 001       : Polarity is positive.(Out of Hysteresis area)
    else if ( <IASTS> = 111)                            : The previous polarity was negative.
        if ( IA ≥ |HYS| )           <IASTS> = 001       : The polarity is changed to positive.
        else if ( IA > HYS )        <IASTS> = 101       : The polarity is changed to positive.(Hysteresis area)
    else if ( <IASTS> = 011)
        if ( IA ≤ −|HYS| )          <IASTS> = 111       : Negative (Out of Hysteresis area)
( Calculates the other 2-phase in the same manner.)
```

	Register name	Function	
Input	VExHYS	Hysteresis for current polarity determination	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExDCTS	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> xx0: Undefined, x01: Positive current, x11: Negative current
	VExMODE[15]	Current polarity determination control	<IPDEN> 0: Determination is disabled. 1: Determination is enabled.
	VExTMPREG0	a-phase current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase current	
	VExTMPREG2	c-phase current	
Output	VExDCTS	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> x0: Undefined, 01: Positive current, 11: Negative current

2. Input process 2 (Task 10)

Input process 2 task supports 3-shunt current detection (3-phase detection and 2-phase detection) and 2-sensor current detection. It also supports 1-shunt current detection when PWM outputting in PWM Shift 2 mode. A current direction can be specified in each phase.

Note: The zero-current detection mode is not available in input process 2 task.

Note: PWM Shift can only be selected in 1-shunt current detection mode.

- Input conversion

<Equation>

[VDC fixed-point transformation/store]		
if (VExMODE[4] = 0	VExVDC = [DC voltage] >>1	
else	VExVDCL = [DC voltage] >>1	
[Current 1 read]		
if ([Current 1 phase information] = 1)	VExIAADC = [Current 1]	
else if ([Current 1 phase information] = 2)	VExIBADC = [Current 1]	
else if ([Current 1 phase information] = 3)	VExICADC = [Current 1]	
[Current 2 read]		
N = [Current 2 phase information] X = [Current 2]		: Detected by 1 ADC sequentially.
if (VExFMODE[13] = 1	N = [Current 4 phase information] X = [Current 4]	: Current 4 is used at synchronous sampling.
if (N = 1)	VExIAADC = X	
else if (N = 2)	VExIBADC = X	
else if (N = 3)	VExICADC = X	
[Current 3 read]		
if (VExFMODE[3:2] = 00)		: Only when 3-phase detection is performed.
if ([Current 3 phase information] = 1)	VExIAADC = [Current 3]	
else if ([Current 3 phase information] = 2)	VExIBADC = [Current 3]	
else if ([Current 3 phase information] = 3)	VExICADC = [Current 3]	
else	N = 6 − N − [Current 1 phase information]	: Current 3 is calculated except when 3-phase detection is performed.
[Current fixed-point transformation/store]		
IA = VExIAO − VExIAADC		
if (VExFMODE[5] = 1)	IA = −IA	: Ia current detection setting
IB = VExIBO − VExIBADC		
if (VExFMODE[6] = 1)	IB = −IB	: Ib current detection setting
IC = VExICO − VExICADC		
if (VExFMODE[7] = 1)	IC = −IC	: Ic current detection setting
if (VExFMODE[3:2] ≠ 00)		: Current 3 is calculated except when 3-phase detection is performed.
if (N = 1)	IA = − IB − IC	
else if (N = 2)	IB = − IC − IA	
else if (N = 3)	IC = − IA − IB	

VExTMPREG0 = IA
VExTMPREG1 = IB
VExTMPREG2 = IC

	Register name	Function	
Input	(DC voltage)	Inputs the result of AD conversion.	16-bit data (The conversion result is stored in the upper 12 bits.)
	(Current 1)		
	(Current 2)		
	(Current 3)		
	(Current 4)		
	VExMODE[4]	VDC store register	<VDCSEL>
	VExFMODE[3:2]	Current detection mode	<IDMODE>
	VExFMODE[7:5]	Polarity of the current detection	<ICPLMD>,<IBPLMD>,<IAPLMD> 0: Shunt mode(In = VExInO − VExInADC) 1: Sensor mode(In = VExInADC − VExInO) Note: n = A, B, or C
	VExFMODE[13]	Synchronous sampling control	<SADCEN> 0: Synchronous sampling is disabled. 1: Synchronous sampling is enabled.
Output	VExVDC	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExVDCL	DC supply voltage	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExIAADC	The result of current conversion of a-phase	16-bit data (The result is stored in the upper 12 bits.)
	VExIBADC	The result of current conversion of b-phase	
	VExICADC	The result of current conversion of c-phase	
	VExTMPREG0	a-phase current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase current	
	VExTMPREG2	c-phase current	
Internal	VExIAO	The result of zero-current conversion of a-phase	16-bit data (The result is stored in the upper 12 bits.)
	VExIBO	The result of zero-current conversion of b-phase	
	VExICO	The result of zero-current conversion of c-phase	

- Current polarity determination

<Equation>

```
if ( VExMODE[15] = 1)                                  : Enables the polarity determination.
    IA = VExTMPREG0
    if ( <IASTS> = xx0)                                : Polarity is undefined.
        if ( IA ≥ |HYS| )          <IASTS> = 001       : The polarity is positive.
        else if ( IA ≤ −|HYS| )    <IASTS> = 111       : The polarity is negative.
    else if ( <IASTS> = 001)                           : The previous polarity is positive.
        if ( IA ≤ −|HYS| )         <IASTS> = 111       : The polarity is changed to negative.
        else if ( IA < −HYS )      <IASTS> = 011       : The polarity is changed to negative.(Hysteresis area)
    else if ( <IASTS> = 101)
        if ( IA ≥ |HYS| )          <IASTS> = 001       : The polarity is positive.(Out of Hysteresis area)
    else if ( <IASTS> = 111)                           : The previous polarity is negative.
        if ( IA ≥ |HYS| )          <IASTS> = 001       : The polarity is changed to positive.
        else if ( IA > HYS )       <IASTS> = 101       : The polarity is changed to positive.(Hysteresis area)
    else if ( <IASTS> = 011)
        if ( IA ≤ −|HYS| )         <IASTS> = 111       : The polarity is negative.(Out of Hysteresis area)
( Calculates the other 2-phase in the same manner.)
```

	Register name	Function	
Input	VExHYS	Hysteresis for current polarity determination	16-bit fixed-point data (0.0 to 1.0, 15 fractional bits)
	VExDCTS	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> x0: Undefined, 01: Positive current, 11: Negative current
	VExMODE[15]	Current polarity determination control	<IPDEN> 0: Determination is disabled. 1: Determination is enabled.
	VExTMPREG0	a-phase current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase current	
	VExTMPREG2	c-phase current	
Output	VExDCTS	Dead time compensation control/status	<ICSTS>, <IBSTS>, <IASTS> xx0: Undefined, x01: Positive current, x11: Negative current

18.4.2.7 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 based on Ia, Ib, and Ic.

   <Equation>

```
if ( VExFMODE[12] = 0 )                                                  : Enables phase transformation.
    VExTMPREG3 = VExTMPREG0                                              : Calculates Iα.
    VExTMPREG4 = 1 ÷ sqrt(3) × VExTMPREG1 − 1 ÷ sqrt(3) × VExTMPREG2     : Calculates Iβ.
else if ( VExFMODE[12] = 1 )                                             : Disables phase transformation.
    VExTMPREG3 = VExTMPREG0
    VExTMPREG4 = VExTMPREG1
```

	Register name	Function	
Input	VExTMPREG0	a-phase current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG1	b-phase current	
	VExTMPREG2	c-phase current	
	VExFMODE[12]	Phase conversion is disabled.	<PHCVDIS>
Output	VExTMPREG3	α-axis current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis current	

2. Input coordinate axis transformation (Task 4)

   In input coordinate axis transformation task, the values of Id and Iq are calculated from Iα, Iβ, VExSINM, and VExCOSM.

   a. Coordinate axis transformation

      <Equation>

```
if ( VExMCTLFM[5] = 0 )                                          : Check the small pulse flag.
    VExID = VExCOSM × VExTMPREG3 + VExSINM × VExTMPREG4          : Calculates the value of Id.
    VExIQ = − VESINM × VExTMPREG3 + VExCOSM × VExTMPREG4         : Calculates the value of Iq
```

	Register name	Function	
Input	VExTMPREG3	α-axis current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExTMPREG4	β-axis current	
	VExSINM	The sine of an angle theta	16-bit fixed-point data (−1.0 to 1.0, 15 fractional bits)
	VExCOSM	The cosine of an angle theta	
	VExMCTLF[5]	Small pulse flag	<PLSLFM>
Output	VExID	d-axis current	32-bit fixed-point data (−1.0 to 1.0, 31 fractional bits)
	VExIQ	q-axis current	

b. ATAN calculation

When <ATANMD> = "10" , ATAN task calculates the declination angle of d-axis current.

When <ATANMD> = "11", ATAN task calculates the induced voltage of d-axis and q-axis. Then it calculates the declination angle using motor voltage equation based on these results.

<Equation>

```
if ( VExMODE[6] = 1 )                                          : Enables calculation of the declination angle.
    LD = VExCLD × [VExCLG setting]                             : d-axis inductance
    LQ = VExCLQ × [VExCLG setting]                             : q-axis inductance
    R = VExCR × [VExCRG setting]                               : Resistance
    VDIV = VExVD − (VExID × R − VExOMEGA × VExIQ × LQ)         : Induced voltage for d-axis
    VQIV = VExVQ − (VExIQ × R + VExOMEGA × VExID × LD)         : Induced voltage for q-axis
    if ( VExMODE[5] = 1 )  VExDELTA = ATAN2(VQIV, VDIV)       : Calculates the declination angle of the induced voltage.
    else                   VExDELTA = ATAN2(VExIQ, VExID)     : Calculates the declination angle of the current.
```

	Register name	Function	
Input	VExCLD	Motor d-axis inductance	16-bit fixed-point data (11 fractional bits)
	VExCLQ	Motor q-axis inductance	
	VExCR	Motor resistance value	
	VExCLG	Inductance range setting	000: One time, 001: $1/2^4$-fold, 010: $1/2^8$-fold, 011: $1/2^{12}$-fold 100: $1/2^{16}$-fold, 101 to 111: Reserved
	VExCRG	Resistor range setting	
	VExOMEGA	Rotational speed	16-bit fixed-point data (15 fractional bits)
	VExVD	d-axis voltage	32-bit fixed-point data (−1.0 to 1.0 and 31fractional bits)
	VExVQ	q-axis voltage	
	VExMODE[6:5]	ATAN operation mode setting	<ATANMD> 0x: Calculation is prohibited. 10: Calculates the declination angle for Id and Iq. 11: Calculates the declination angle of induced voltage of d-axis and q-axis.
Output	VExDELTA	Output of the declination angle	16-bit data (−180 to 180, 0x8000 to 0x7FFF)

18.4.2.8 Other Tasks

1. ATNA2 (Arc tangent function 2) (Task 12)

ATAN2 task calculates the angle from the starting point based on X-axis that draws a straight line from the starting point to the point of (X,Y) on the X-Y coordinate.

<Equation>

X = VExTMPREG4
Y = VExTMPREG5
Z = ATAN (|Y| ÷ |X|) : Arc tangent calculation, 0 to 90°
if (X < 0 & Y ≥ 0) Z = 0x00008000 − Z : The second quadrant (90 to 180°)
if (X < 0 & Y < 0) Z = 0xFFFF8000 + Z : The third quadrant (−90 to −180°)
if (X ≥ 0 & Y < 0) Z = − Z : The forth quadrant (0 to −90°)
if (X = Y = 0) Z = 0x00000000 : Output on the original point (0°)
VExTMPREG5 = Z

	Register name	Function	
Input	VExTMPREG4	Input X	32-bit signed data
	VExTMPREG5	Input Y	
Output	VExTMPREG5	The value of the phase	32-bit data (0xFFFF8000 to 0x00008000 (−180to 180°))

2. SQRT (Square root function) (Task 13)

SQRT task outputs the value from 0.0 to 2.0 based on the calculation of the square root function using the inputs from 0.0 to 4.0.

<Equation>

X = VExTMPREG5 : Inputs (0 to 4.0)
N = 0
if (X < 0x2000) N = 1
if (X < 0x0800) N = 2
if (X < 0x0200) N = 3
if (X < 0x0080) N = 4
if (X < 0x0020) N = 5
if (X < 0x0008) N = 6
if (X < 0x0002) N = 7
if (X ≥ 0x8000) N = −1
$X = X \times 2^{2N}$: Normalization (0.25 to 1.0)
if (X > 0x7FFF) X = 0x7FFF
Z = SQRT(X) : Calculates the square root of X.
if (X = 0) Z = 0 Output is the range of 0.5 to 1.0.
$Z = Z \div 2^{N}$: Inverted transformation (0 to 2.0)
VExTMPREG5 = Z

	Register name	Function	
Input	VExTMPREG5	Input value	Indicates the data on 32-bit fixed-point format. (0.0 to 4.0, 15 fractional bits) 0x0000_0000 to 0x0001_FFFF
Output	VExTMPREG5	The value of square root	32-bit fixed-point data (0.0 to 2.0, 15 fractional bits) 0x0000_0000 to 0x0000_FFFF

18.5 Combinations between the VE channel, the PMD and the ADC

Usable combinations between the vector engine, the PMD, and the ADC are restricted depending on the channel of the vector engine.

Note that this combination changes depending on the method of the current detection.

In 3-shunt and 2-sensor current detection, synchronous sampling can be performed using two ADCs. In this case, current inputs to be used are Current 1 and Current 4.

Table 18-8 Combinations between the vector engine and the PMD

Vector engine	PMD channel 0	PMD channel 1
Channel 0	o	−
Channel 1	−	o

Table 18-9 Combinations between the vector engine and the ADC (without synchronous sampling)

Vector engine		ADC unit A				ADC unit B			
Channel	Current detection VExMODE <IDMODE[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	00	Current 1	Current 2	Current 3	DC voltage	−	−	−	−
	01	Current 1	Current 2	−	DC voltage	−	−	−	−
	1x	Current 1	Current 2	−	DC voltage	−	−	−	−
1	00	−	−	−	−	Current 2	Current 1	Current 3	DC voltage
	01	−	−	−	−	Current 2	Current 1	−	DC voltage
	1x	−	−	−	−	Current 2	Current 1	−	DC voltage

Table 18-10 Combinations between the vector engine and the ADC (with synchronous sampling)

Vector engine		ADC unit A				ADC unit B			
Channel	Current detection VExMODE <IDMODE[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	00	Current 1	−	Current 3	DC voltage	Current 4	−	−	−
	01	Current 1	−	−	DC voltage	Current 4	−	−	−
	1x	−	−	−	−	−	−	−	−
1	00	−	Current 4	−	−	−	Current 1	Current 3	DC voltage
	01	−	Current 4	−	−	−	Current 1	−	DC voltage
	1x	−	−	−	−	−	−	−	−

19. Encoder Input Circuit (A-ENC)

19.1 Outline

The encoder input circuit (A-ENC) supports 6 operation modes: encoder mode, sensor mode (3 types), timer mode, and phase counter mode.

The A-ENC has the following functions:

- Supports incremental type encoders (Z-phase is available) and Hall sensor ICs.
- Rotational edge detection (2-phase input: 4-fold, 3-phase input: 6-fold)
- Rotational direction detection circuit
- Counter function (event count, timer count, and phase count functions)
- 2 types of interrupt outputs
- Interrupt request output enable/disable signals and the event flags
- Incorporates the digital-noise filter in the input circuit
- Sampling can be performed synchronizing with PWM signals in the PMD.
- Zero-cross detection (PMD circuit is used) for BEMF (induced voltage) of BLDC motors that operate on rectangular wave drive.
- Usable as a general-purpose 32-bit timer/capture.

19.2 Block Diagram

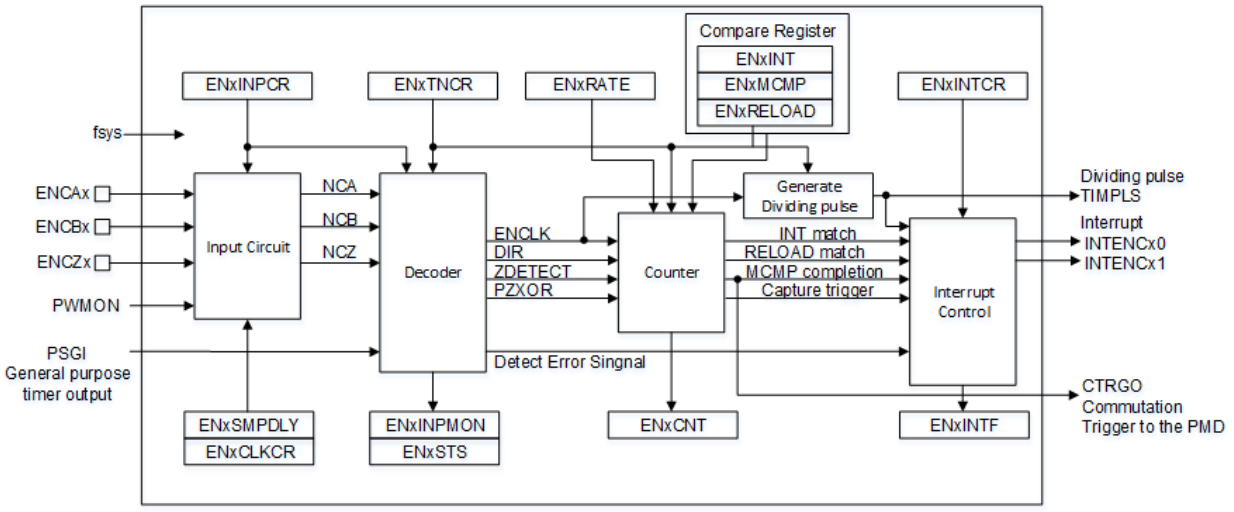


Figure 19-1 Block diagram of encoder input circuit

19.3 Registers

19.3.1 Register List

The following table lists the control registers and their addresses:

For the base address, refer to "A list of peripheral function base addresses" in the chapter on "Memory Map."

Register name		address (Base+)
ENC control register	ENxTNCR	0x0000
RELOAD compare register	ENxRELOAD	0x0004
INTcompare register	ENxINT	0x0008
Counter/capture register	ENxCNT	0x000C
MCMP compare register	ENxMCMP	0x0010
Phase count rate register	ENxRATE	0x0014
Status register	ENxSTS	0x0018
Input process control register	ENxINPCR	0x001C
Sample delay register	ENxSMPDLY	0x0020
Input monitor register	ENxINPMON	0x0024
Sample clock control register	ENxCLKCR	0x0028
Interrupt request control register	ENxINTCR	0x002C
Interrupt event flag register	ENxINTF	0x0030

19.3.2 ENxTNCR(ENC Control Register)

	31	30	29	28	27	26	25	24
Bit symbol	-	-	-	CMPSEL	UDMD		TOVMD	MCMPMD
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit symbol	DECMD		SDTEN	-	MODE			P3EN
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	TRGCAPMD	SFTCAP	ENCLR	ZESEL	
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	ZEN	ENRUN	-	-	-	ENDEV		
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-29	−	R	Read as "0".
28	CMPSEL	R/W	Counter clear condition in timer mode. 0: A match of the ENxINT register 1: A match of the ENxRELOAD register
27-26	UDMD	R/W	Sets the up/down count setting in sensor mode (phase count) or phase count mode. 00: Up count 01: Down count 1x: Sets to up or down count setting with the ENxRATE register When "1x" is set to this bit, if ENxRATE<RATE> < 0, the counter is set to down count setting. If <RATE> ≥ 0, the counter is set to up count setting.
25	TOVMD	R/W	Sets the counter operation when the RELOAD register matches the value of the counter (timeout operation setting). [Sensor mode (timer count)] 0: Counting continues. 1: Counting stops. When the counter stops, the match should be cleared by software to restart the counter. [Timer mode, sensor mode (phase count), phase counter mode] 0: The counter is cleared and counting continues. 1: Counting stops When the counter stops, the match should be cleared by software to restart the counter. [Encoder mode] Regardless of the TOVMD setting, when the motor drives on CW direction, the counter is cleared and counting continues, when the motor drives on CCW direction, counting continues. [Sensor mode (event count)] This bit is not used for a match with the RELOAD register.
24	MCMPMD	R/W	[Sensor mode (timer count), or timer mode] Compare mode for the ENxMCMP register. 0: Compare match (ENxMCMP<MCMP> = counter value 1: Comparison in size (ENxMCMP<MCMP> ≤ counter value Sets to <MCMPMD>=0 except in sensor mode (timer count) or timer mode.

Bit	Bit symbol	Type	Function
23-22	DECMD	R/W	[Encoder mode or sensor mode] Sets the detection direction for the decoder. 00: CW or CCW edge detection Detects the variation of input signals (ENCAx, ENCBx, and ENCZx). 01: CW edge detection Detects the variation of the input signal from the previous state on rotational edge detection to the current state. (The result of detection is maintained.) 10: CCW edge detection Detects the variation of the input signal from the previous state on rotational edge detection to the current state. (The result of detection is maintained.) 11: CW or CCW edge detection Detects the variation of the input signal from the previous state on rotational edge detection to the current state. (The result of detection is maintained.) Sets to <DECMD>=00 in timer mode or phase count mode.
21	SDTEN	R/W	Sets skip detection at 3-phase decoding in sensor mode. 0: Detection is disabled. 1: Detection is enabled. When a skip is detected, an error flag (ENxSTS<SKPDT>) is set.
20	−	R	Read as "0".
19-17	MODE[2:0]	R/W	Sets the operation mode. 000: Encoder mode 001: Sensor mode (event count) 010: Sensor mode (timer count) 011: Timer mode 100: Reserved 101: Reserved 110: Sensor mode (phase count) 111:Phase counter mode
16	P3EN	R/W	[Sensor mode](Note 1) Sets the decoding mode (2-phase/3-phase input selection) 0: 2-phase decoding 1: 3-phase decoding
15-13	−	R	Read as "0".
12	TRGCAPMD	R/W	[Sensor mode (timer count or phase count), timer mode, or phase counter mode] Trigger capture operation selection 0: Capturing and clearing the counter 1: Only capturing Selects the capture operation at rotational edge detection in sensor mode (timer count or phase count). Selects the capture operation when the z-input is enabled in timer mode or phase counter mode. When software capturing is used, the counter is not cleared.
11	SFTCAP	W	[Sensor mode (timer count, phase count), timer mode, or phase counter mode] Performs software capturing. 1: Captures the value of the counter. When this bit to "1", the value of the counter is captured. To obtain the captured value, read the ENxCNT register. Writing "0" has no meaning. Read as "0".
10	ENCLR	W	Clears the counter. 1: Clear When this bit to "1", the counter is cleared to "0". After the counter is cleared, the counter restarts operation. Writing "0" has no meaning. Read as "0".
9-8	ZESEL	R/W	[Timer mode or phase counter mode] Selects the detection edge when the Z-phase input is enabled. (The Z-phase input/the PSGI input) 00: Reserved 01: A rising edge is detected. 10: A falling edge is detected. 11: Both edges are detected.

Bit	Bit symbol	Type	Function
7	ZEN	R/W	[Encoder mode, Timer mode, or phase counter mode] (Note 2) Enables/disables Z-phase input. 0: The ENCZ input is disabled. 1: The ENCZ input is enabled.
6	ENRUN	R/W	Enables/disables the encoder input circuit. 0: Disabled 1: Enabled When <ENRUN>="1", <ZDET> is cleared to "0" and the encoder input circuit is also enabled. When <ENRUN>="0", the encoder input circuit is disabled.
5-3	−	R	Read as "0".
2-0	ENDEV[2:0]	R/W	[Encoder mode, or sensor mode (event count)] Sets the ratio of a divided output (TIMPLS) of a rotational edge pulse. According to this setting, a rotational edge pulse is divided and the this pulse is used as an interrupt event. 000: divided by 1 100: divided by 16 001: divided by 2 101: divided by 32 010: divided by 4 110: divided by 64 011: divided by 8 111: divided by 128

Note 1: Set <P3EN> to "0" except in sensor mode.

Note 2: Set <ZEN> to "0" except in encoder mode, timer mode, and phase counter mode.

The operation mode is determined with <MODE[2:0]>, <P3EN>, and <ZEN>. They are 12 types in total.

The following table lists the operation mode settings:

<MODE[2:0]>	<ZEN>	<P3EN>	Input pin	Mode
000	0	0	A, B	Encoder mode
	1		A, B, Z	Encoder mode (use of the Z-input)
001	0	0	A, B	Sensor mode (event count)
		1	A, B, Z	Sensor mode (event count and use of the Z-input)
010	0	0	A, B	Sensor mode (timer count and the 2-phase input)
		1	A, B, Z	Sensor mode (timer count and the 3-phase input)
011	0	0	-	Timer mode
	1		Z	Timer mode (use of the Z-input)
110	0	0	A, B	Sensor mode (phase count and the 2-phase input)
		1	A, B, Z	Sensor mode (phase count and the 3-phase input)
111	0	0	-	Phase count mode
	1		Z	Phase count mode (use of the Z-input)

19.3.3 ENxRELOAD (RELOAD Compare Register)

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

Bit	Bit symbol	Type	Function
31-16	RELOADH	R/W	[Encoder mode] Sets the maximum value of the counter. Sets the number of input pulses per rotation (after × 4 − 1). [Sensor mode (phase count) or phase counter mode] Sets the maximum value (count range per rotation) of the counter. [Sensor mode (timer count) or timer mode] This bit is used as the register to compare with the counter. If a match occurs, an interrupt occurs. Sets the upper 16 bits of 32 bits for comparison. [Sensor mode (event count)] This bit is not used.
15-0	RELOADL	R/W	[Sensor mode (timer count) or timer mode] Sets the lower 16 bits for 32-bit comparison. [Others] This bit is not used.

19.3.4 ENxINT(INT Compare Register)

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

Bit	Bit symbol	Type	Function
31-16	INTH	R/W	[Encoder mode or sensor mode (event count)] Sets the value on 16 bits for comparison with the counter. When the counter matches the value of <INTH>, an INT match signal occurs. At this time, an interrupt can be generated. [Sensor mode (timer count) or timer mode] Sets the value on 32 bits for comparison with the counter. Sets the upper 16 bits of 32 bits to <INTH> for comparison. When the counter matches the value of <INTH>, an INT match signal occurs. At this time, an interrupt can be generated. This match signal can be used as a position detection start signal when the PWM synchronous sampling is enabled. [Sensor mode (phase count) or phase counter mode] Sets the value on 16 bits for comparison with the counter. When the counter matches the value of <INTH>, an INT match signal occurs. At this time, an interrupt can be generated.
15-0	INTL	R/W	[Sensor mode (timer count) or timer mode] Sets the lower 16 bits for 32-bit comparison. [Others] This bit is not used.

19.3.5 ENxCNT (Counter Register)

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

Bit	Bit symbol	Type	Function
31-16	CNTH	R	[Encoder mode or sensor mode (event count)] Reads the count value of a rotational edge pulse. [Sensor mode (timer count or phase count)] Reads the captured value or software captured value at rotational edge detection/at the ENCZx input edge detection. [Timer mode or phase counter mode] Reads the captured value or software captured value at the ENCZ input edge detection. [Sensor mode (timer count) or timer mode] Reads the value of the upper 16 bits at 32-bit capturing.
15-0	CNTL	R	[Encoder mode, sensor mode (event count or phase count), or phase count mode] This bit is not used. [Sensor mode (timer count) or timer mode] Reads the value of the lower 16 bits at 32-bit capturing.

19.3.6 ENxMCMP (MCMP Counter Register)

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

Bit	Bit symbol	Type	Function
31-16	MCMPH	R/W	[Sensor mode (Timer count) or timer mode] Sets the value on 32 bits for comparison with the counter. Sets the upper 16 bits of 32 bit to <MCMPH> for comparison. An interrupt can be generated. Comparison in size (<MCMPMD>=1) An MCMP completion signal is output when ENxMCMP ≤ a counter value is established. In this mode, one completion signal is output in each time the register is written. Compare match mode (<MCMPMD>=0) An MCMP completion signal is output when ENxMCMP = a counter value is established. [Except sensor mode (timer count) or timer mode] Sets the value on 16 bits for comparison with the counter. An interrupt can be generated. An MCMP completion signal is output when ENxMCMP<MCMPH> = a counter value. When ENxINTCR<MCMPIE> is enabled, a commutation trigger signal is output to the PMD circuit.
15-0	MCMPL	R/W	[Sensor mode (timer count) or timer mode] Sets the lower 16 bit of 32 bits for comparison. [Others] This bit is not used.

19.3.7 ENxRATE (Phase Count 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	RATE[15:8]							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	RATE[7:0]							
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-15	−	R	Read as "0".
15-0	RATE	R/W	[Sensor mode (phase count) or phase counter mode] Set the count frequency of the counter. Clock frequency generation: fsys × <RATE>/2^{16} Depending on the ENxTNCR<UDMD> setting, the value of <RATE> can be specified as a signed bit or unsigned bit. If the value of <RATE> is negative, the counter counts down. <UDMD>=0x: Unsigned, 0 or more/less than 1.0 (0x0000 to 0xFFFF) <UDMD>=1x: Signed, -0.5 or more/less than 0.5 (0x8000 to 0x7FFF) When <UDMD>=1x, set two's complement to <RATE>.

19.3.8 ENxSTS (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	-	REVERR	UD	ZDET	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	SKPDT	PDERR	INERR
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-15	−	R	Read as "0".
14	REVERR	R	[Sensor mode (timer count or phase count)] Sets the inverted flag of <UD> when the both edges are detected. (Note 1)(Note 2) 0: - 1: An inversion of <UD> is detected. If ENxTNCR<ENRUN> = "0", "0" is always set.
13	UD	R	[Encoder mode or sensor mode (event count, timer count, or phase count)] Sets the motor rotation direction. 0: CCW (Counterclockwise) 1: CW (Clockwise) When the motor rotates to CW direction, this bit is "1". When the motor rotates to CCW direction, this bit is "0". If ENxTNCR<ENRUN> = "0", "0" is always set.
12	ZDET	R	Detects passing of the ENCZ input. 0: A Z-input is not detected after the encoder input has been enabled. 1: A Z-input is detected. This bit is cleared when ENxTNCR<ENRUN>="0".
11-3	−	R	Read as "0".
2	SKPDT	R	Detects a skip detection flag when skip detection is enabled. (Note 1) 0: Undetected 1: A skip is detected.
1	PDERR	R	[Encoder mode, sensor mode (event count, timer count, or phase count)] Detects an decoding detection error flag. (Note 1) 0: Undetected 1: An error is detected.
0	INERR	R	[Sensor mode (event count, timer count, or phase count)] Detects an abnormal input error. (Note 1) 0: No abnormal error 1: Abnormal error This bit is set to "1" when all 3-phase inputs on 3-phase decoding are "Low" or "High".

Note 1: The flag is cleared by reading this register.

Note 2: After the mode transition, make sure to read a flag first and then clear to "0".

19.3.9 ENxINPCR (Input Process 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	-	NCT						
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	PDSTP	PDSTT	-	-	-	SYNCNCZEN	SYNCSPLMD	SYNCSPLEN
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-15	−	R	Read as "0".
14-8	NCT	R/W	Sets the noise cancel time. Setting range: 0 to127 (0x00 to 0x7F) Cancel time: Setting value × Sample clock cycle (with the <SPLCKS> setting) When this bit is set to "0", noise cancelling does not operate. The sampling clock in PWM-off edge sample mode is a PWM signal.
7	PDSTP	W	[Sensor mode (timer count or phase count)] Sets the position detection stop command at PWM synchronous sampling. (BFMF detection control) 1: Stops position detection When this bit is set to "1", position detection is started. Writing "0" has no meaning. Read as "0".
6	PDSTT	W	[Sensor mode (timer count or phase count)] Sets the position detection start command at PWM synchronous sampling. (BEMF detection control) 1: Starts position detection When this bit is set to "1", position detection is started. Writing "0" has no meaning. Read as "0".
5-3	−	R	Read as "0".
2	SYNCNCZEN	R/W	Controls a noise cancel counter when sampling is performed in the PWM-on period. 0: The counter stops in the PWM-off period. 1: The counter stops and cleared in the PWM-off period. This setting is used to enable the PWM synchronous sampling (<SYNCSPLEN>=1) and to select the PWM sampling in the PWM-on period (<SYNCSPLMD> =0).
1	SYNCSPLMD	R/W	Selects the PWM synchronous sampling. 0: Sampling in the PWM-on period 1: Sampling in the PWM-off edge This setting is enabled when the PWM synchrony sampling is enabled (<SYNCSPLEN>=1).
0	SYNCSPLEN	R/W	Enables the PWM synchronous sampling 0: Continuous sampling 1: PWM synchronous sampling This bit enables the PMD synchronous sampling. When the PWM synchronous sampling is set in sensor mode (timer count or phase count), if <SYNCSPLEN> is set to "1", BEMF detection control is enabled on decoding operation.

19.3.10 ENxSMPDLY (Sample Delay Register)

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

Bit	Bit symbol	Type	Function
31-8	−	R	Read as "0".
7-0	SMPDLY	R/W	Sampling disable period Setting range: 0 to 255 (0x00 to 0xFF) Disabled time: <SMPDLY> setting value × Sampling cycle (with the <SPLCKS> setting) Sets the sampling disable period after the PWM-on edge in the PWM-on period (ENxINPCR<SYNCSPLEN>=1, ENxINPCR<SYNCSPLMD>=0) has elapsed.

19.3.11 ENxINPMON (Input 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	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit symbol	-	DETMONZ	DETMONB	DETMONA	-	SPLMONZ	SPLMONB	SPLMONA
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-7	−	R	Read as "0".
6	DETMONZ	R	Monitors the position detection status for NCZ. Stores the value of NCZ at position detection.
5	DETMONB	R	Monitors the position detection status for NCB. Stores the value of NCB at position detection.
4	DETMONA	R	Monitors the position detection status for NCA. Stores the value of NCA at position detection.
3	−	R	Read as "0".
2	SPLMONZ	R	Monitors the status of ENCZx after noise cancelling. Stores the status of NCZ that is the ENCZx input after noise cancelling.
1	SPLMONB	R	Monitors the status of ENCBx after noise cancelling. Stores the status of NCB that is the ENCBx input after noise cancelling.
0	SPLMDNA	R	Monitors the status of ENCAx after noise cancelling. Stores the status of NCA that is the ENCAx input after noise cancelling.

19.3.12 ENxCLKCR (Sample Clock 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	-	-	-	-	-	-	SPLCLK	
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-2	−	R	Read as "0".
1-0	SPLCKS	R/W	Sets the sampling frequency. 00: fsys 01: fsys/2 10: fsys/4 11: fsys/8 Sets the sampling frequency for the ENCAx input, the ENCBx input, and the ENCZx input. On PWM synchronous sampling, this bit is disabled, when off edge sampling is set (ENxINPCR<SYNCSPLEN>=1 and ENxINPCR <SYNCSPLMD>=1).

19.3.13 ENxINTCR (Interrupt 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	-	-	MCMPIE	RLDIE	CMPIE	ERRIE	CAPIE	TPLSIE
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-6	−	R	Read as "0".
5	MCMPIE	R/W	Enables/disables a MCMP completion interrupt. 0: Disabled 1: Enabled When this bit to "1", an INTENCx1 occurs at MCMP completion.
4	RLDIE	R/W	Enables/disables a RELOAD interrupt. 0: Disabled 1: Enabled When this bit to "1", an INTENCx1 occurs when the counter matches the value of the RELOAD register. A RELOAD interrupt.does not occur in encoder mode or sensor mode (event count).
3	CMPIE	R/W	Enables/disables a INT interrupt 0: Disabled 1: Enabled When this bit to "1", an INTENCx1 occurs when the counter matches the value of the INT register.
2	ERRIE	R/W	Enables/disable a detection error interrupt. 0: Disabled 1: Enabled When this bit to "1", an edge detection error (PDERR) occurs, or when a skip is detected (SKPDT), an INTENCx0 occurs. An interrupt does not occur in timer mode or phase counter mode.
1	CAPIE	R/W	Enables/disable a capture trigger interrupt. 0: Disabled 1: Enabled When this bit to "1", if the value of the counter is captured due to the external trigger (an ENCZx input) or the rotational edge pulse (ENCLK), an INTENCx0 occurs. An interrupt does not occur in encoder mode or sensor mode (event count).
0	TPLSIE	R/W	Enables/disable a rotational divided edge interrupt. 0: Disabled 1: Enabled When this bit to "1", an INTENCx0 occurs by rotational edge divided pulses. An interrupt does occurs only in encoder mode and sensor mode (event count).

19.3.14 ENxINTF (Interrupt Event 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	-	-	MCMPF	RLDCPF	INTCPF	ERRF	CAPF	TPLSF
After reset	0	0	0	0	0	0	0	0

Bit	Bit symbol	Type	Function
31-6	−	R	Read as "0".
5	MCMPF	R	MCMP compare completion flag 0: No flag 1: The flag is generated.
4	RLDCPF	R	RELOAD match flag 0: No flag 1: The flag is generated. This bit is not set in encoder mode or sensor mode (event count).
3	INTCPF	R	INT match flag 0: No flag 1: The flag is generated.
2	ERRF	R	Detection error flag 0: No flag 1: The flag is generated. This bit is not set in timer mode or phase counter mode.
1	CAPF	R	Capture flag 0: No flag 1: The flag is generated. This bit is not set by soft capture. This bit is not set in encoder mode or sensor mode (event count).
0	TPLSF	R	Rotational edge divided pulse flag 0: No flag 1: The flag is generated. This bit is enabled in encoder mode or sensor mode (event count).

Note: Each flag is set by the occurrence of the event that was enabled, and it is cleared by reading the ENxINTF register.

19.4 Operation Description

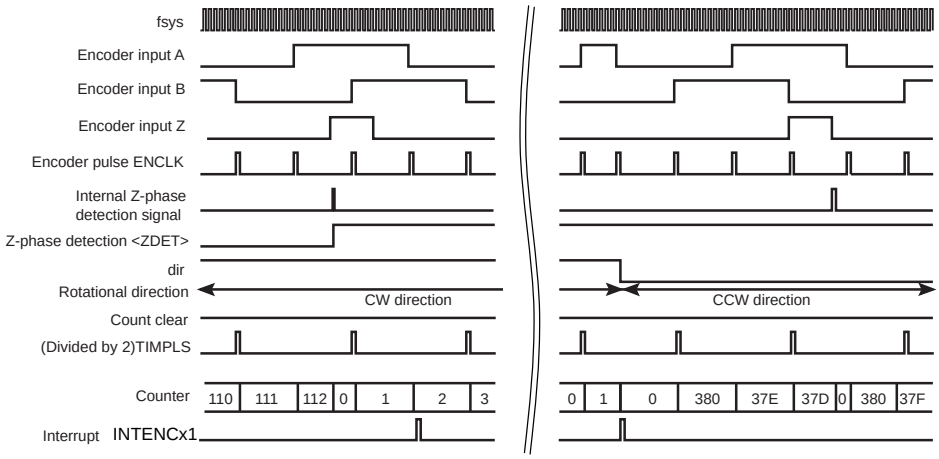
19.4.1 Encoder Mode

The A-ENC supports the high-speed position sensor (phase determination) that determines the phase input from the incremental encoder (AB encoder or ABZ encoder).

- Performs rotational edge detection, and then outputs the divided pulses and interrupt requests.
- Generates an interrupt request on rotational edge pulse count or the specified count value.
- Determines the rotational direction.
- up/down count (controlled by the rotational direction)
- Sets the range of counts.
- Sets the rotational direction for the detection.
- An abnormal error detection flag is available.

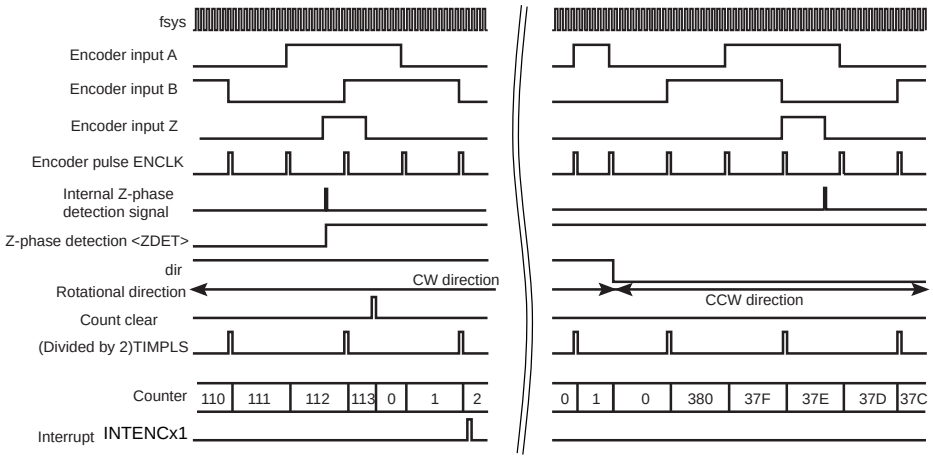
1. The ENCZ input is enabled. (ENxTNCR<ZEN> = 1)

   ENxRELOAD<RELOADH[15:0]> = 0x0380, ENxINT<INTH[15:0]> = 0x0002



2. The ENCZ input is disabled. (ENxTNCR<ZEN> = 0)

   ENxRELOAD<RELOADH[15:0]> = 0x0380, ENxINT<INTH[15:0]> = 0x0002



In encoder mode, the incremental encoder inputs should be connected to the ENCA, ENCB, and ENCZ pins. The encoder multiplies the ENCA and the ENCB signals into 4 to count the encoder pulses.

During the CW rotation (i.e., A-phase has the 90-degree phase lead to B-phase), the counter counts up; when the value of the counter reaches the value of <RELOAD>, the counter is cleared to "0" on the next ENCLK.

During the CCW rotation (i.e., A-phase has the 90-degree phase lag to B), the counter counts down; when the value of the counter reaches "0x0000", the value of <RELOAD> is set to the counter on the next ENCLK.

Additionally, when <ZEN> = "1", the counter is cleared to "0" on the rising edge of Z-phase during the CW rotation and on the falling edge of Z-phase during the CCW rotation. If the ENCLK matches Z-phase detection timing, the encoder counter is cleared to "0" without increment or decrement.

When "1" is written to ENxTNCR<ENCLR>, the counter is cleared to "0".

ENxSTS<UD> is set to "1" during the CW rotation and is cleared to "0" during the CCW rotation.

The detection direction (CW direction or CCW direction) can be set with ENxTNCR<DECMD[1:0]>. Except when <DECMD>="00", a rotational edge is detected based on the comparison between the input status ENxINPMON<DETMONA><DETMONB><DETMONZ> at the previously detected edge and the current input value.

Then the signal (TIMPLS) of which the ENCLK signal is divided is output.

When ENxINTCR<CMPIE> = "1", if the value of <ENINTH> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<MCMPIE> = "1", the value of ENxMCMP<MCMPH[15:0]> and the counter value become equal an INTENCx1 interrupt occurs.

However, when <ZEN> = "1", a match interrupt does not occur during ENxSTS<ZDET> = "0". After the encoder input is enabled, if the encoder counter detects the first Z-signal, <ZDET> is set to "1".

<ZDET> and ENxSTS<UD> are cleared to "0" when ENxTNCR<ENRUN> = "0".

19.4.2 Sensor Mode

The counter supports low-speed position sensing (zero-cross determination) to deal 2-phase Hall sensor input and 3-phase Hall sensor input. There are three mode: event count mode, timer count mode, and phase count mode.

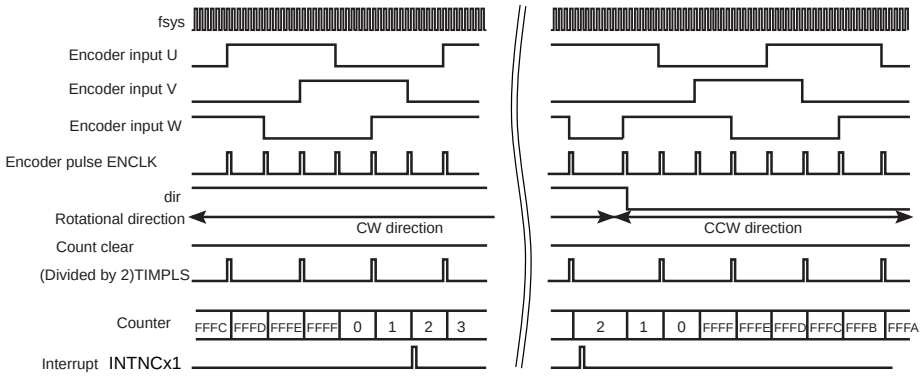
In timer count mode or phase count mode, when a brushless DC motor is driven on rectangular waveforms from the PMD circuit, the A-ENC can perform zero-cross detection of an induced voltage using PWM synchronous sampling.

19.4.2.1 Event Count

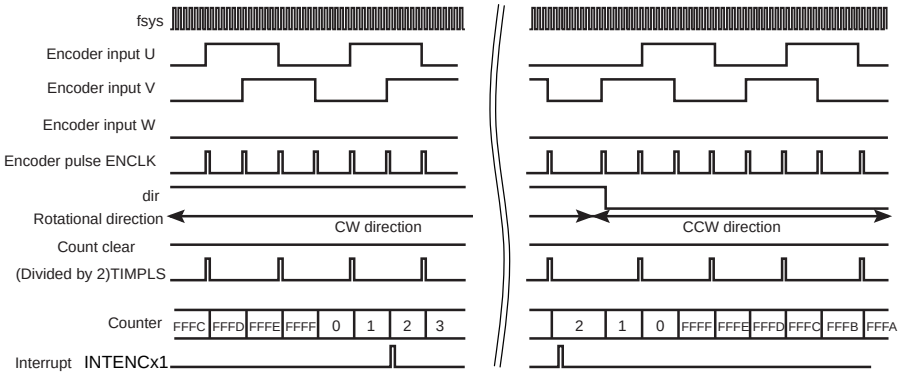
The counter counts using rotational edge detection.

- Performs rotational edge detection and outputs a divided pulse and interrupt request.
- Generates an interrupt request on rotational edge pulse count or the specified count value.
- Determines the rotational direction.
- Sets up- or down count (controls by the determination of rotational direction).
- Specifies the rotational detection direction.
- Abnormal error detection flag

1. 3-phase decoding (ENxTNCR<P3EN> = 1)
   ENxINT<INTH[15:0]> = 0x0002)



2. 2-phase decoding (ENxTNCR<P3EN> = 0)
   ENxINT<INTH[15:0]> = 0x0002)



Hall sensor inputs (U,V, and W) should be connected to ENCAx, ENCBx, and ENCZx. When <P3EN> = "0", the counter multiplies a 2-phase input (ENCAx or ENCBx) by 4; when <P3EN> = "1", the counter multiplies a 3-phase input (ENCAx, ENCBx, or ENCZx) by 6. After that, the counter counts the Hall sensor pulse.

During the CW rotation (i.e., ENCA has a 90-degree phase lead to ENCB channel), the counter performs up counting. When the value of the counter reaches "0xFFFF", the encoder counter is cleared to "0" on the next ENCLK.

During the CCW rotation (i.e., ENCA has a 90-degree phase lag to ENCB), the encoder counter performs down counting. When the value of the counter reaches "0x0000", the counter is set to "0xFFFF" on the next ENCLK.

When "1" is written to ENxTNCR<ENCLR>, the counter is cleared to "0".

When the CW rotation is detected, ENxSTS<UD> is set to "1". When the CCW rotation is detected, ENxSTS<UD> is cleared to "0".

The detection direction (CW direction or CCW direction) can be set with ENxTNCR<DECMD[1:0]>. Except when <DECMD>="00", a rotational edge is detected based on the comparison between the input status ENxINPMON <DETMONA><DETMONB><DETMONZ> at the previously detected edge and the current input value.

Then the signal (TIMPLS) of which the ENCLK signal is divided is output.

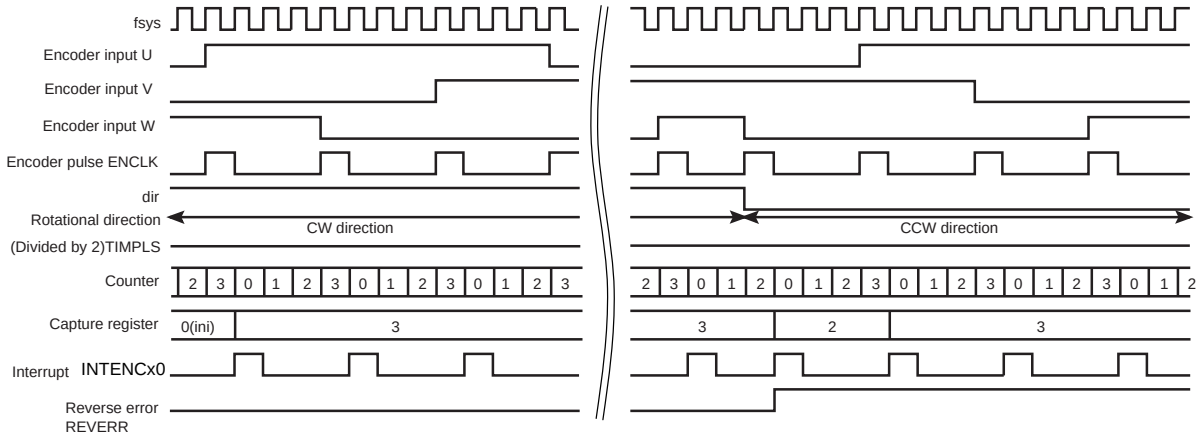
When ENxINTCR<CMPIE> = "1", if the value of <ENINTH> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<MCMPIE> = "1", if the value of <ENMCMPH> and the counter value become equal, an INTENCx1 interrupt occurs.

19.4.2.2 Timer Count

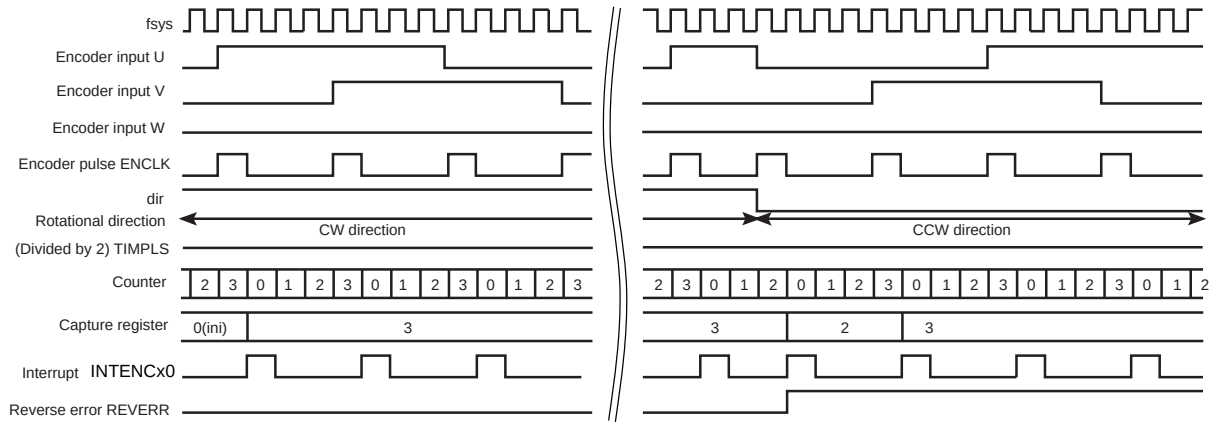
1. 3-phase decoding (ENxTNCR<P3EN> = 1)

   ENxINT<INTH[15:0]><INTL[15:0]> = 0x00000002



2. 2-phase decoding (ENxTNCR<P3EN> = 0)

   ENxINT<INTH[15:0]><INTL[15:0]> = 0x00000002



Hall sensor inputs (U,V, and W) should be connected to ENCAx, ENCBx, and ENCZx. When <P3EN> = "0", the counter multiplies a 2-phase input (ENCAx or ENCBx) by 4; when <P3EN> = "1", the counter multiplies a 3-phase input (ENCAx, ENCBx, or ENCZx) by 6. After that, the counter generates a 4-fold or 6-fold rotational edge pulse (ENCLK).

The counter always performs up counting and is cleared to "0x00000000" on ENCLK.

When "1" is written to ENxTNCR<ENCLR>, the counter is cleared to "0x00000000".

The counter value is captured on ENCLK. The captured value can be read from the ENxCNT register.

When "1" is written to ENxTNCR<SFTCAP>, the counter value is captured. The timing of capturing can be specified by the user. The captured value can be read from the ENxCNT register.

The value of the ENxCNT register (captured value) is maintained regardless of the value of ENxTNCR<ENRUN>.

When the CW rotation is detected, ENxSTS<UD> is set to "1". When the CCW rotation is detected, ENxSTS<UD> is cleared to "0".

When <ENRUN> = "0", <UD> is cleared to "0". If the rotational direction is reversed, ENxSTS<REVERR> is set to "1". This flag is cleared upon reading.

The detection direction (CW direction or CCW direction) can be set with ENxTNCR<DECMD[1:0]>.

Except when <DECMD>="00", a rotational edge is detected based on the comparison between the input status ENxINPMON <DETMONA><DETMONB><DETMONZ> at the previously detected edge and the current input value.

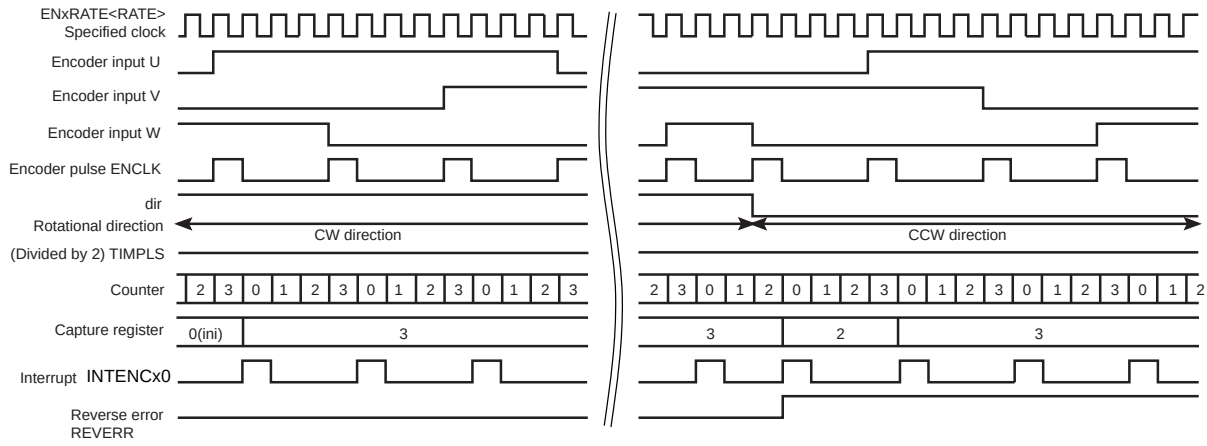
When ENxINTCR<RLDIE>="1", if the value of ENxRELOAD<RELOADH[15:0]><RELOADL[15:0]> and the counter val-ue become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<CMPIE> = "1", if the value of ENxINT<INTH[15:0]><INTL[15:0]> and the counter value be-come equal, an INTENCx1 interrupt occurs.

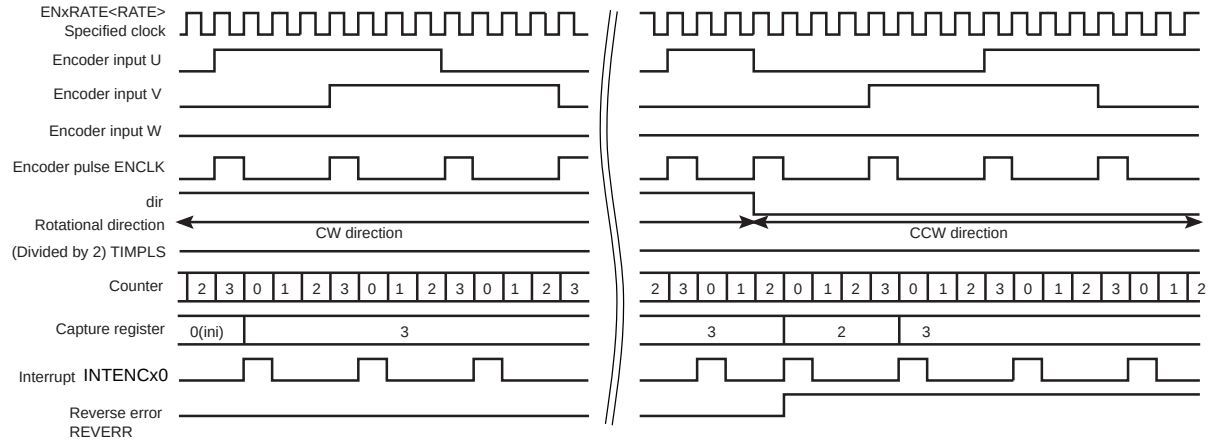
When ENxINTCR<MCMPIE> = "1", if the value of ENxMCMP<MCMPH[15:0]><MCMPL[15:0]> and the counter val-ue become equal, an INTENCx1 interrupt occurs. When ENxINTCR<MCMPMD> ="1", if the counter val-ue is the value of <MCMPH> or more, an INTENCx1 interrupt occurs.

19.4.2.3 Phase Count

1. 3-phase decoding (ENxTNCR<P3EN> = 1)



2. 2-phase decoding (ENxTNCR<P3EN> = 0)



Hall sensor inputs (U,V, and W) should be connected to ENCAx, ENCBx, and ENCZx. When <P3EN> = "0", the counter multiplies a 2-phase input (ENCAx or ENCBx) by 4; when <P3EN> = "1", the counter multiplies a 3-phase input (ENCAx, ENCBx, or ENCZx) by 6. After that, the counter generates a 4-fold or 6-fold rotational edge pulse (ENCLK).

The counter can be set to up/down count by the <UDMD> setting or the RATE register setting at the specified frequency. When the counter is set to up count, if the value of the counter and the value of <RELOADH> become equal, the counter is cleared to "0". When the counter is set to down count, if the value of the counter reaches "0x0000", the value of <RELOADH> is set to the counter.

When "1" is written to <ENCLR>, the counter is cleared to "0".

When <TOVMD> ="1", the counter stops at the value of <RELOADH>.

The counter value is captured on ENCLK. The captured value can be read from the ENxCNT register.

When "1" is written to <SFTCAP>, the counter value is captured. The timing of capturing can be specified by the user. The captured value can be read from the ENxCNT register.

The value of the ENxCNT register (captured value) is maintained regardless of the value of <ENRUN>.

When the CW rotation is detected, <UD> is set to "1". When the CCW rotation is detected, <UD> is cleared to "0". When <ENRUN> = "0", <UD> is cleared to "0". If the rotational direction is reversed, ENxSTS<REVERR> is set to "1". This flag is cleared upon reading.

The detection direction (CW direction or CCW direction) can be set with ENxTNCR<DECMD[1:0]>. Except when <DECMD>="00", a rotational edge is detected based on the comparison between the input status ENxINPMON <DETMONA><DETMONB><DETMONZ> at the previously detected edge and the current input value.

When ENxINTCR<CMPIE> = "1", if the value of <ENxINT> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<MCMPIE> = "1" , if the value of <ENxMCMP> and the counter value become equal, an INTENCx1 interrupt occurs.

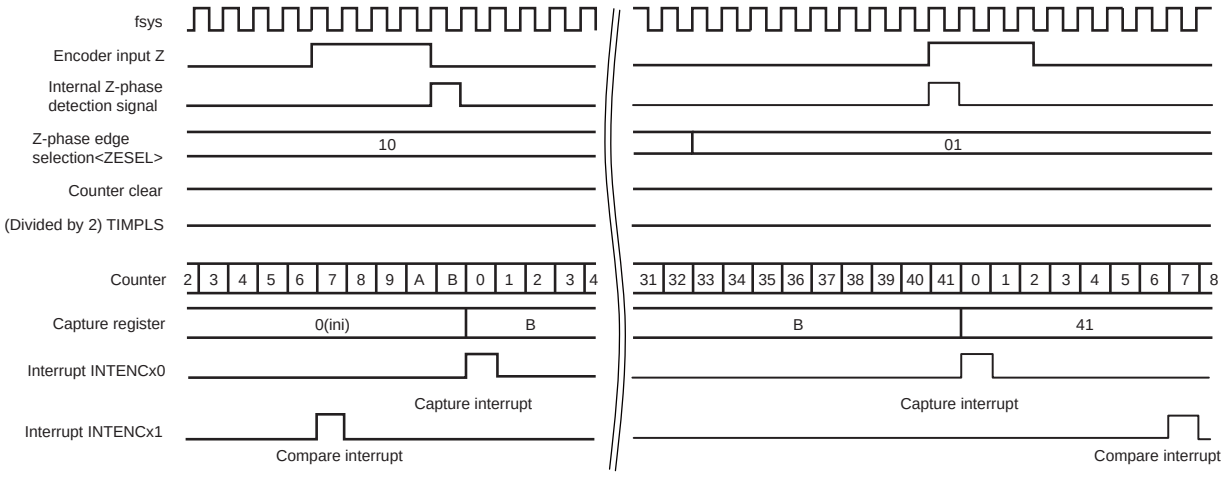
19.4.3 Timer Mode

This mode can be used as a general-purpose 32-bit timer.

- 32-bit up counter (counts using the fsys clock)
- Counter clear control (software clear, compare match clear, and external trigger)
- A match interrupt occurs using the compare function.
- Capture functions : external trigger capture (interrupt can be generated), software capture

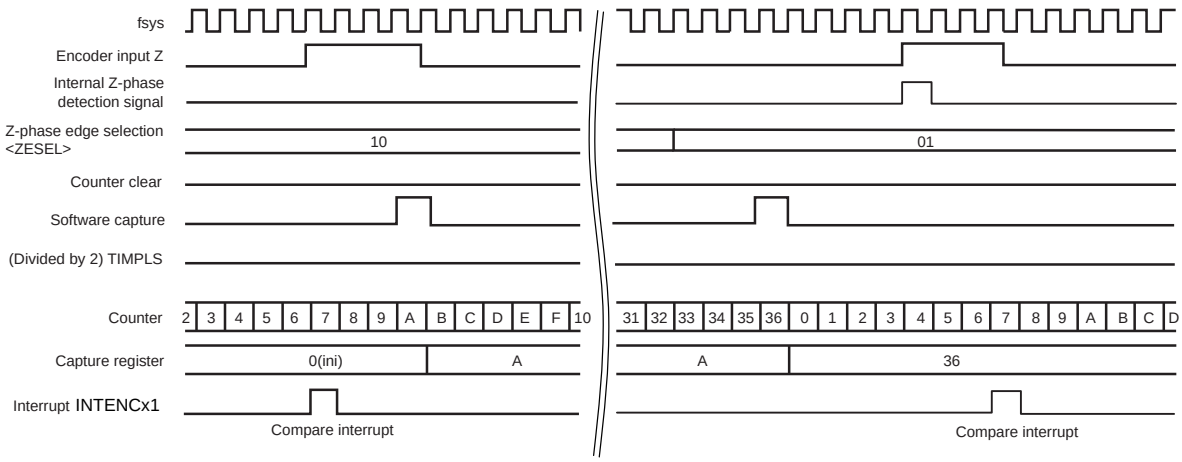
1. The ENCZ input is enabled. (ENxTNCR<ZEN> = 1)

   ENxINT<INTH[15:0]><INTL[15:0]> = 0x0000_0006



2. The ENCZ input is disabled.(ENxTNCR<ZEN> = 0)

   ENxINT<INTH[15:0]><INTL[15:0]> = 0x0000_0006)



When <ZEN> = "1", a Z-input can work as the external trigger. When <ZEN> = "0", the external trigger is not used.

The counter always performs up counting.

When "1" is written to ENxTNCR<ENCLR>, the counter is cleared to "0".

When <ZEN> = "1", if ENxTNCR<ZESEL> is set to "01", the counter is cleared on the rising edge of the Z-phase. When <ZESEL> = "10", the counter is cleared on the falling edge of the Z-phase. When <ZESEL> ="11", the counter is cleared on the both edge of the Z-phase.

When a Z-phase edge is detected, the counter value is captured. The captured value can be read from the ENxCNT register.

When "1" is written to ENxTNCR<SFTCAP>, the counter value is captured. The timing of capturing can be specified by the user. The captured value can be read from the ENxCNT register.

The value of the ENxCNT register (captured value) is maintained regardless of the value of ENxTNCR<ENRUN>. The clearing event of the captured value is only a reset.

When ENxINTCR<RLDIE>="1", the value of ENxRELOAD<RELOADH[15:0]><RELOADL[15:0]> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<CMPIE> = "1", the value of ENxINT<INTH[15:0]><INTL[15:0]> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<MCMPIE> = "1", the value of ENxMCMP<MCMPH[15:0]><MCMPL[15:0]> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxTNCR<MCMPMD> ="1", if the counter value is <MCMPH[15:0]><MCMPL[15:0]> or more, an INTENCx1 interrupt occurs.

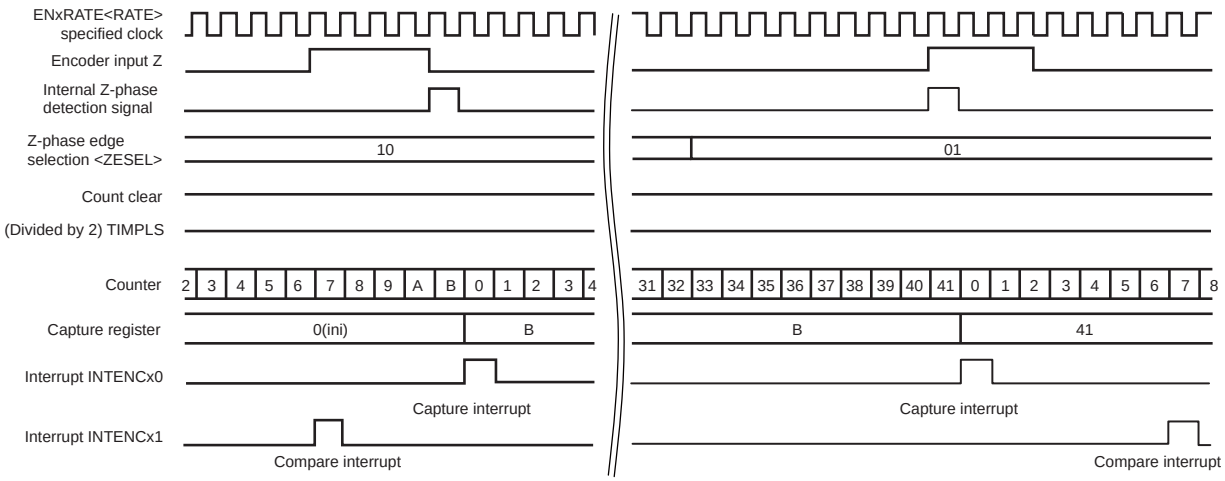
19.4.4 Phase Counter Mode

This mode can be used as a 16-bit counter that can be controlled at the specified frequency.

- Up/down count control is available.
- Sets the compare function and match interrupt request
- Clears the counter/capture function using a ENCZ input, and then occurs an interrupt request

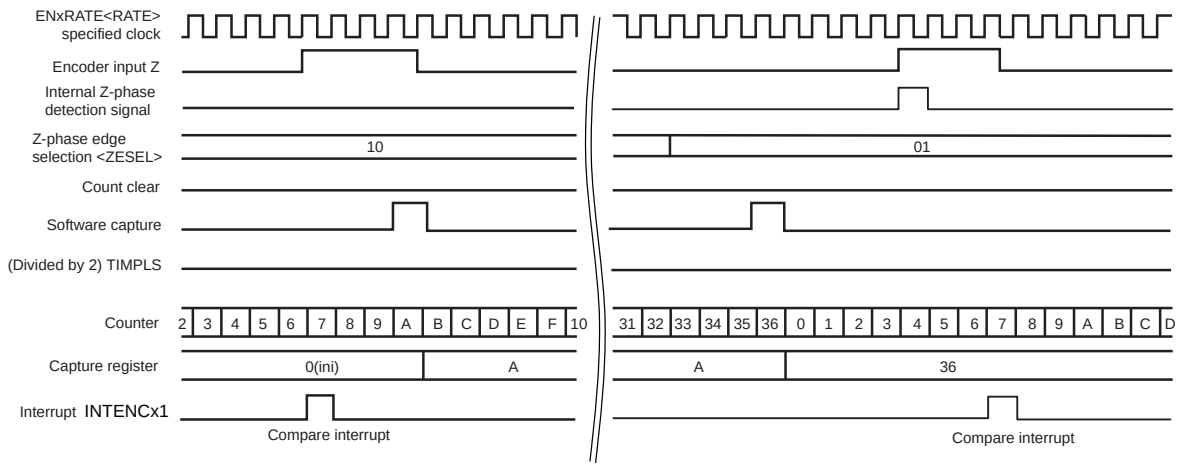
1. The ENCZ is enabled. (ENxTNCR<ZEN> = 1)

ENxINT<INTH[15:0]> = 0x0006



2. The ENCZ input is disabled.(ENxTNCR<ZEN> = 0)

ENxINT<INTH[15:0]> = 0x0006)



When <ZEN> = "1", a Z-input can work as the external trigger. When <ZEN> = "0", the external trigger is not used.

The counter controls up/down counting with the ENxTNCR<UDMD> and the ENxRATE registers at the specified frequency.

When the counter performs up counting, if the value of the counter becomes equal to the value of ENxRELOAD<RELOADH[15:0]>, the counter is cleared to "0".

When the counter preforms down counting, if the value of the counter reaches "0x0000", the value of <RELOADH> is set to the counter.

When ENxTNCR<TOVMD> = "1", the counter stops at the value of <RELOADH>.

When "1" is written to ENxTNCR<ENCLR>, the counter is cleared to "0".

When <ZEN> = "1", if ENxTNCR<ZESEL> = "01", the counter is cleared to "0" on the rising edge of the Z-phase; if <ZESEL> = "10" , the counter is cleared to "0" on the falling edge of the Z-phase; if <ZESEL> ="11", the counter is cleared to "0" on the both edges.

When an edge of the Z-phase is detected, the counter value is captured. The captured value can be read from the ENxCNT register.

When "1" is written to ENxTNCR<SFTCAP>, the counter value is captured. Capturing can be performed at the specified timing. The captured valued can be read from the ENxCNT register.

The value of the ENxCNT register (captured value) is maintained regardless of the value of ENxTNCR<ENRUN>. The clearing event of the captured value is only a reset.

When ENxINTCR<CMPIE> = "1", if the value of ENxINT<INTH> and the counter value become equal, an INTENCx1 interrupt occurs.

When ENxINTCR<MCMPIE> = "1", if the value of ENxMCMP<MCMPH> and the counter value become equal, an INTENCx1 interrupt occurs.

19.4.4.1 Phase Difference Count Mode

In phase counter mode, when <P3EN> = <ZEN> = "1", the counter enters phase-difference counter mode. In this mode, the up/down counter is controlled with the TMRB output and the ENCZ input.

- When the TMRB output and the ENCZ input are equal, the counter operation is set to up counting. When the TMRB output and the ENCZ input are not equal, the counter operation is set to down counting.
- The value of the counter is captured on the edge of the TMRB output. At this time the counter can be cleared and an interrupt request can be generated.

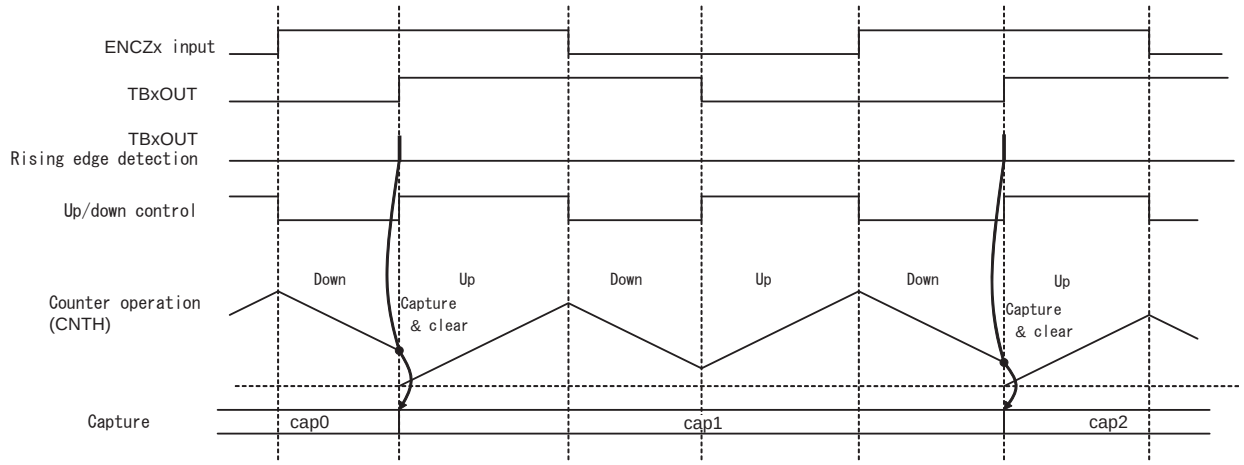


Figure 19-2 Operation in phase counter mode (phase difference)

When an edge of the TBxOUT signal is detected, the captured value and the counter are cleared. At this time, the detection edge is set with ENxTNCR<ZESEL>.

The captured value indicates the phase difference between the ENCZ input and the TBxOUT signal. The standard value (capture value is "0") is assumed that the phase difference between the ENCZ input and the TBxOUT signal is 1/4 cycle.

19.5 Operation Description in Each Circuit

19.5.1 Input Circuit

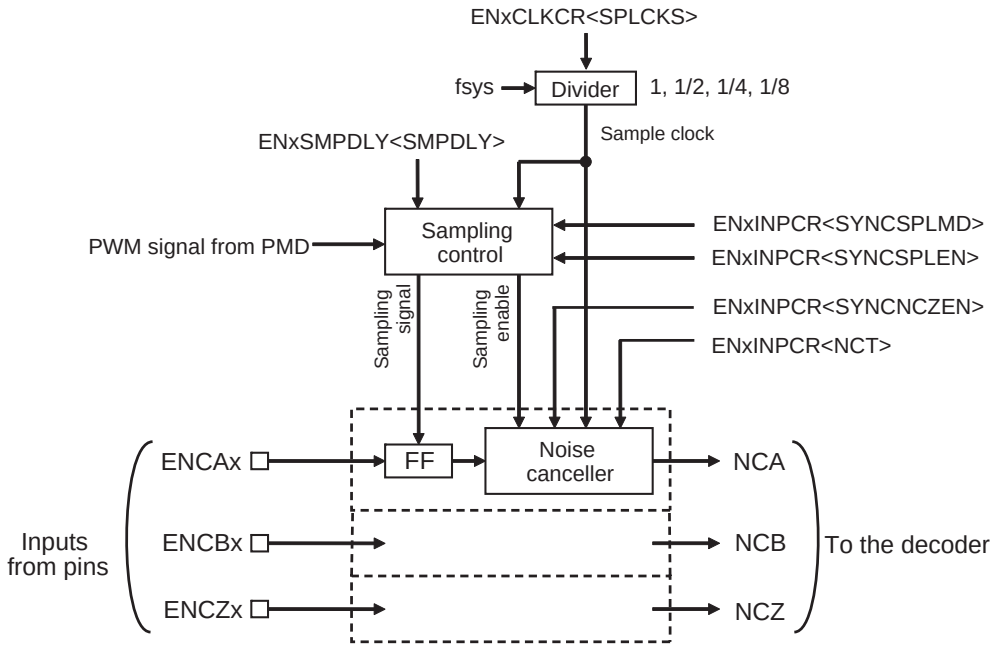


Figure 19-3 Configuration of input circuit

The input circuit samples the inputs from the pins (ENA,ENB,ENZ) on the specified sampling signal to reduce the noise using the digital noise-filter.

19.5.1.1 Sample Clock

The sample clock can be selected from fsys, fsys/2, fsys/4, or fsys/8 with <SPLCKS>.

19.5.1.2 Sampling Mode

1. Continuous sampling (ENxINPCR<SYNCSPLEN>="0")

   In continuous sampling, the input circuit samples the input clock on sample clocks.

2. PWM synchronous sampling (ENxINPCR<SYNCSPLEN>="1")

   In PWM synchronous sampling, the input circuit samples the input clock synchronously with PWM signals from the PMD circuit.

   - PWM-on period sampling (ENxINPCR<SYNSPLMD> ="0")

     In PWM-on period sampling, the input circuit samples the input signal on the specified sample clock selected with ENxCLKCR<SPLCKS> for the PWM-on period.
   - PWM off-edge sampling (ENxINPCR<SYNSPLMD> ="1")

     A PWM signal serves as a sampling signal, and the input circuit samples the input signal on the off-edge of the PWM signal.
   - PWM on-delay setting

The PWM on-delay setting can be set with ENxSMPLDSY<SMPDLY> when the input circuit samples an on-delay period.

Delay time= <SMPDLY> × Sample clock cycle

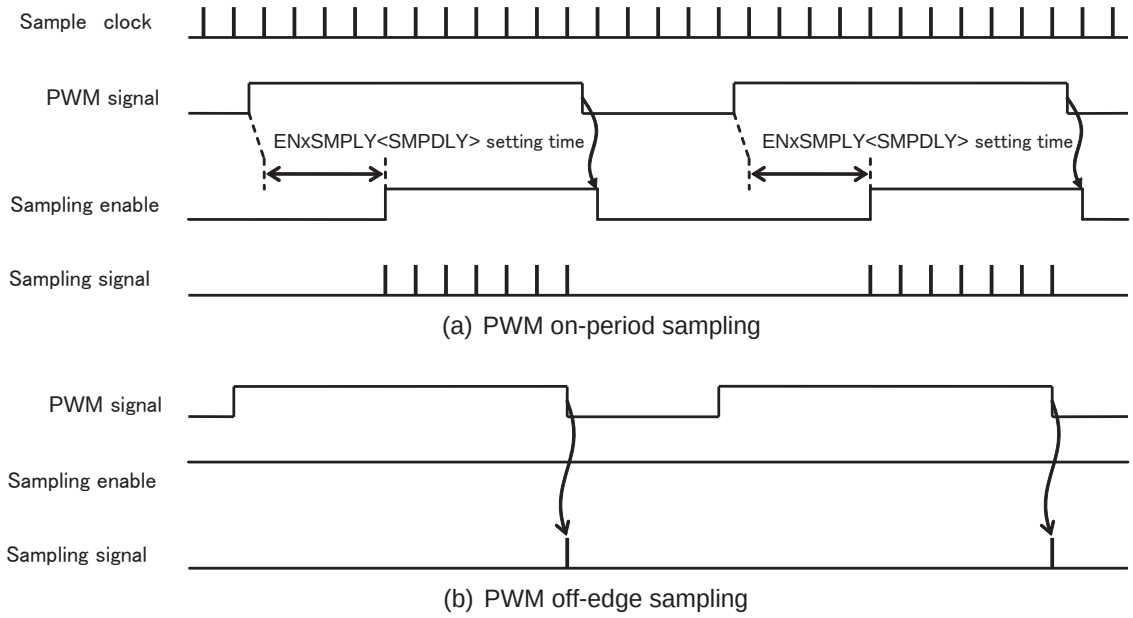


Figure 19-4 PWN synchronous sampling

19.5.1.3 Noise Cancelling

1. Noise cancelling time

   A noise cancelling time can be set with ENxINPCR<NCT[6:0]>. A noise cancelling time can be calculated as follows:

   Noise cancelling time= <NCT> × Sample clock cycle

Note: When "0" is set to <NCT>, noise cancelling is disabled.

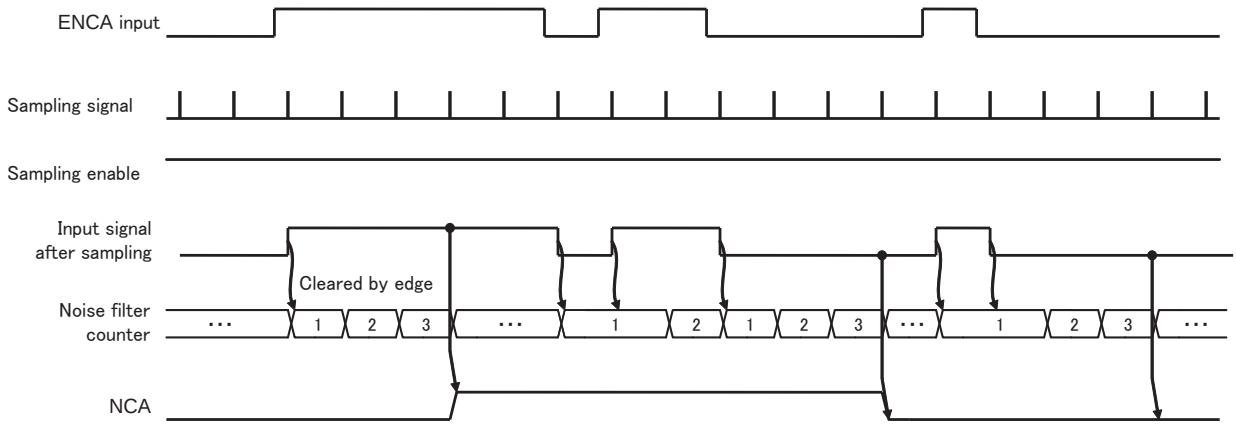


Figure 19-5 Noise cancelling (Continuous, <NCT>=3)

2. Noise cancelling when the input circuit samples the input signal in the PWM-on period.

- Noise cancelling timer stops in the PWM-off period (ENxINPCR<SYNCNCZEN > ="0")
- Noise cancelling timer is cleared in the PWM-off period (ENxINPCR<SYNCNCZEN> ="1")

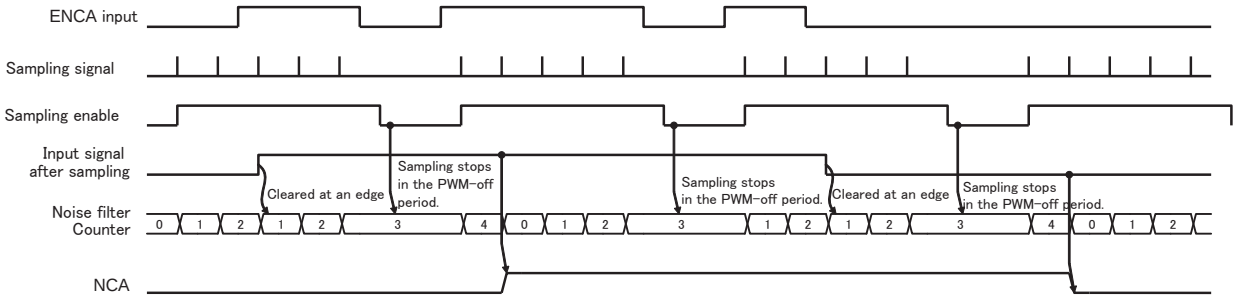


Figure 19-6 Noise cancelling (Sampling in the PWM-on period. Noise cancelling timer stops in the PWM-off period. <NCT>=4)

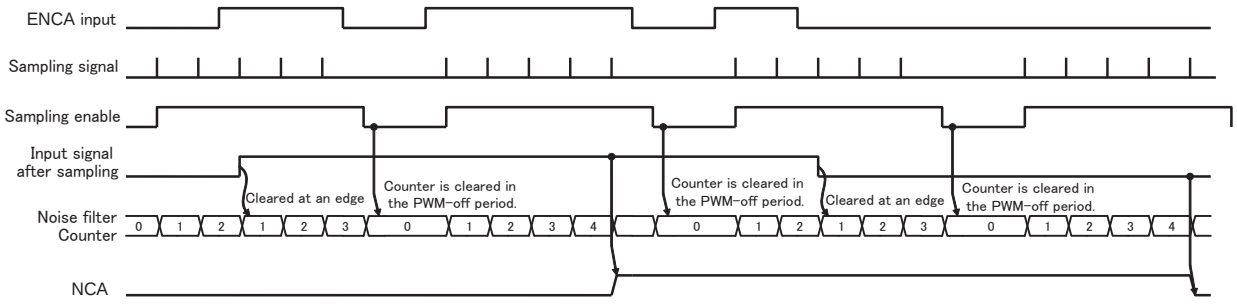


Figure 19-7 Noise cancelling (Sampling in the PWM-on period. Noise cancelling timer is cleared in the PWM-off period.<NCT>=4)

19.5.2 Decoder

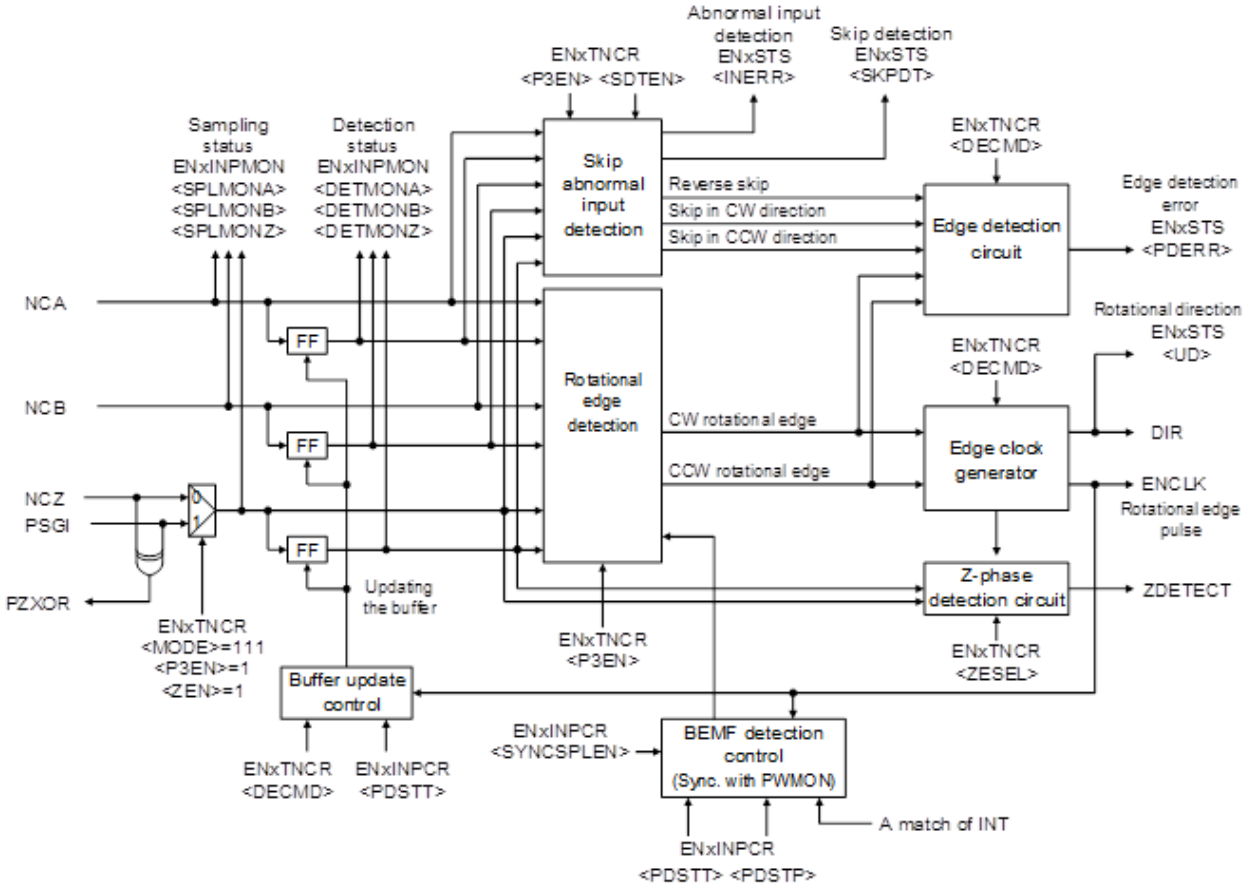


Figure 19-8 Configuration of the decoder

The decoder detects a rotational edge and rotational direction from 2-phase or 3-phase inputs after noise cancelling. It also detects the Z-phase when incremental encoder is connected; it detects an edge of the Z-signal when single-phase is input.

19.5.2.1 Rotational Edge Detection and Direction Signal Generation

1. 2-phase decoding (ENxTNCR<P3EN> ="0")

   Supports encoder mode and sensor mode (by setting <P3EN>="0").

   In 2-phase decoding, four input pattern variations (rotational edge) are detected.

   For the inputs in CW direction, when a rotational edge of (1)→(2), (2)→(3), (3)→(4), or (4)→(1) is detected, the status of ENxSTS<UD> is "1".

   For the inputs in CCW direction, when a rotational edge of (4)→(3), (3)→(2), (2)→(1), or (1)→(4) is detected, the status of ENxSTS<UD> is "0".

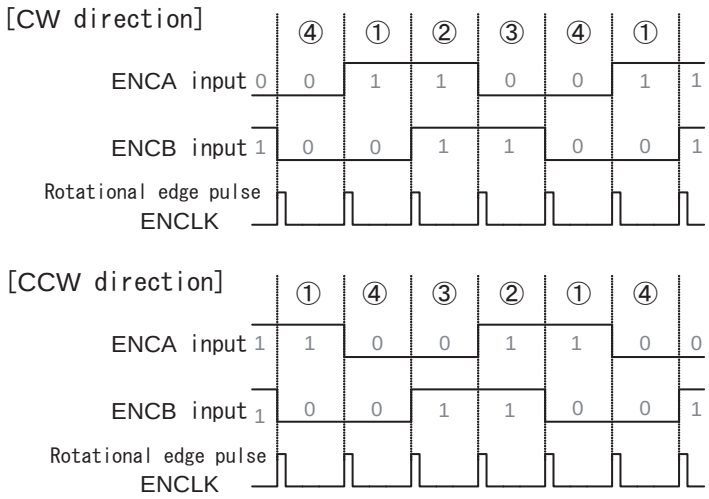


Figure 19-9 Waveforms of 2-phase decoding

2. 3-phase decoding (ENxTNCR<P3EN> ="1")

Available only in sensor mode.

Three-phase decoding detects variations of 6 input patterns (rotational edge).

For inputs in CW direction, when a rotational edge of (1)→(2), (2)→(3), (3)→(4), (4)→(5), (5)→(6), or (6)→(1). the status of ENxSTS<UD> is set to "1".

For inputs in CCW, when a rotational edge of (6)→(5), (5)→(4), (4)→(3), (3)→(2), (2)→(1), or (1)→(6), the status of ENxSTS<UD> is set to "0".

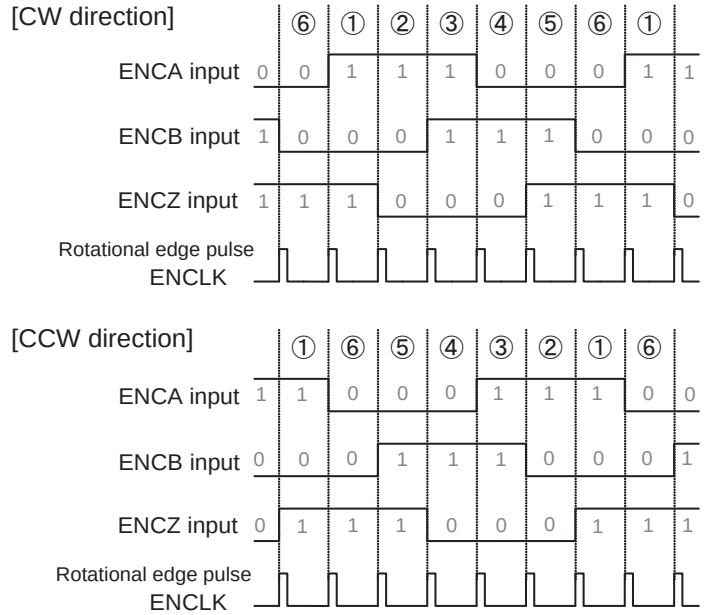


Figure 19-10 Waveforms of 3-phase decoding

19.5.2.2 Z-input Detection Circuit

The decoder detects an edge of the Z-input.

1. Encoder mode

   The decoder detects a rising edge in CW direction and detects a falling edge in CCW direction.

2. Timer mode and phase counter mode

   The decoder selects a detection edge either from a rising edge, falling edge, or both edges with <ZESEL>.

19.5.2.3 Skip Detection and Abnormal Input Detection

1. Skip detection

   Skip detection is enabled when ENxTNCR<SDTEN> ="1".

   If a skip is detected, <SKPDT> is set to "1".

   - Combination of skip detection and 2-phase decoding (ENxSTS<P3EN> ="0")

     Reverse skip detection: (1)→(3), (2)→(4), (3)→(1), (4)→(2)
   - Combination of skip detection and 3-phase decoding (ENxSTS<P3EN> ="1")

     Skip detection in CWdirection: (1)→(3), (2)→(4), (3)→(5), (4)→(6), (5)→(1), (6)→(2)

     Skip detection in CCW direction: (1)→(5), (2)→(6), (3)→(1), (4)→(2), (5)→(3), (6)→(4)

     Reverse skip detection: (1)→(4), (4)→(1), (2)→(5), (5)→(2), (3)→(6), (6)→(3)

2. Abnormal input detection

   In sensor mode (event count, timer count, or phase count), an edge detection of which three inputs are all "0" or all "1" in 3-phase decoding, is detected, the encoder determines these inputs are abnormal. If an abnormal input is detected, ENxSTS<INERR> is set to "1".

19.5.2.4 Edge Detection Error Detection

The encoder determines an error when a direction that is not the specified direction with ENxTNCR<DECMD> is detected. An edge detection error detection is one of the interrupt events.

- Skip detection is disabled. (ENxTNCR<SDTEN> ="0")

  During the CW rotation (ENxTNCR<DECMD> ="01"): an error occurs when a rotational edge of CCW direction is detected.

  During the CCW rotation (ENxTNCR<DECMD> ="10"): an error occurs when a rotational edge of CW direction is detected.
- Skip detection is enabled. (ENxTNCR<SDETEN> ="1")

  During the CW rotation (ENxTNCR<DECMD> ="01"): an error occurs when a skip in CCW direction, a reverse skip, or rotational edge of CCW direction is detected.

  During the CCW rotation (ENxTNCR<DECMD> ="10"): an error occurs when a skip in CW direction, a reverse skip, or rotational edge of CW direction is detected.

19.5.2.5 Buffer Update Control

When ENxTNCR<DECMD> ="00", the buffer is always enabled. In this case, rotational edge detection and skip detection are determined by the variations of input signals.

If <DECMD> is set to other than "00", buffer updating is only performed when a rotational edge is detected. Therefore, the result of the edge detection and skip detection is stored in the buffer. A rotational

edge is detected based on the comparison between the input status (ENxINPMON<DETMONA><DETMONB><DETMONZ>) at the previously detected edge and the current input value (ENxINPMON <SPLMONA><SPLMONB><SPLMONZ>).

19.5.2.6 BEMF Detection Control

In sensor mode (timer count and phase count), BEMF detection control is enabled when ENxINPCR<SYNCSPLEN> ="1" (PWM synchronous sampling). It can stop (suspend) or start (restart) the rotational edge detection.

This control is used when position detection (position sensorless) is performed on the induced voltage of the brushless DC motor that drives on rectangular waveforms generated by the motor control circuit (PMD).

1. Starting the rotational edge detection
   - Command operation: Write "1" to ENxINPCR<PDSTT>
   - Event operation: An INT compare match from the counter circuit

2. Stopping the rotational edge detection
   - Command operation: Write "1" to ENxINPCR<PDSTP>
   - Event operation: A rotational edge is detected.

19.5.3 Counter

The counter circuit consists of the clock generator, counter, compare function, and capture function. Usable functions vary depending on the operation mode.

19.5.3.1 Encoder Mode and Sensor Mode (Event Count)

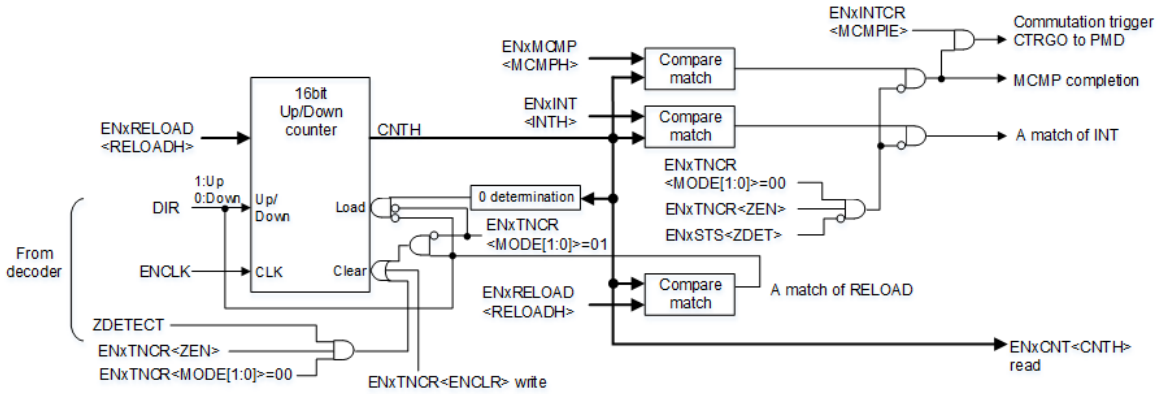


Figure 19-11 Configuration of the counter (encoder mode and sensor mode (event count))

The counter circuits uses a rotational edge pulse (enclk) from the decoder, a 16-bit up/down counter operating on rotational direction signal (dir), and three types of compare functions: <RELOADH>, <INTH>, and <MCMPH>.

In encoder mode, the counter is cleared by a match of RELOAD compare during the CW rotation. When a CCW rotation detection is used, if the counter is determined as "0", the value of the <RELOADH> is loaded to the counter.

In encoder mode, when Z-input is enabled (ENxTNCR<ZEN>="1"), a match signal of INT compare and MCMP compare are ignored during when the encoder input is enabled (ENxTNCR<ENRUN> ="1") to when an edge of the Z-phase is detected.

The value of the up/down counter can be read by the counter register (ENxCNT). When ENxINTCR<MCMPIE> ="1", a MCMP compare match signal serves as a commutation trigger for the PMD circuit.

19.5.3.2 Sensor Mode(Timer count) or Timer mode

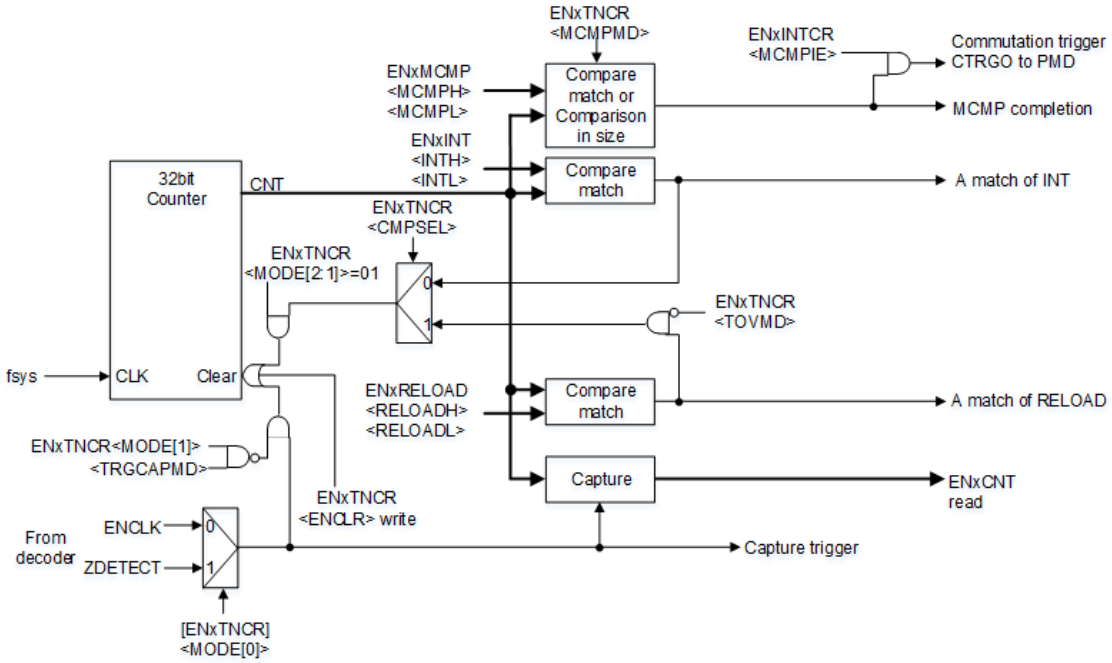


Figure 19-12 Configuration of the counter (sensor mode (timer count) or timer mode)

The counter consists of the 32-bit counter operating on the system clock (fsys), three-types of the compare functions (RELOAD, INT, and MCMP), and the capture function.

The MCMP compare function can select the selection type from match compare and compare in size. When comparison in size is selected (ENxTNCR <MCMPMD>="1"), comparison starts by setting the ENxMCMP register. If the conditions are met, a MCMP completion signal is output and comparison finished.

The counter is cleared by a match of INT compare or RELOAD compare in timer mode.

In sensor mode (timer count), the counter is captured and cleared when a rotational edge is detected (ENCLK). In timer mode, the counter is captured and cleared when a Z-edge is detected (ZDETECT). The captured value can be read by reading the counter register (ENxCNT).

When ENxINTCR<MCMPIE>="1", a MCMP compare match signal serves as a commutation trigger signal for the PMD circuit.

19.5.3.3 Sensor mode (Phase Count) or Phase Counter Mode

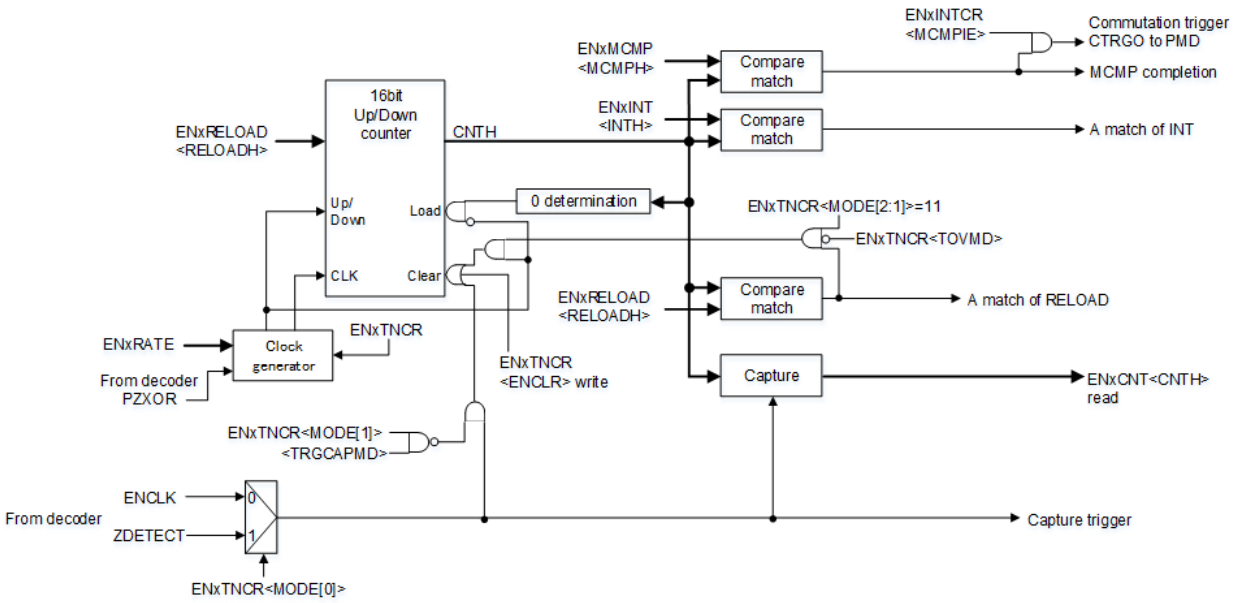


Figure 19-13 Configuration of the counter (sensor mode (phase count), and phase counter mode)

The counter consists of the clock generator specified with ENxRATE<RATE>, a 16-bit up/down counter operating on the clock signal and direction signal from the clock generator, three-types of compare match comparators (<RELOADH>, <INTH>, and <MCMPH>), and capture function.

The counter clock is specified with the ENxRATE register at the specified rate.

The up/down counter is specified with ENxTNCR<UDMD>.

When the counter is set to up counting, the counter is cleared by a match of RELOAD compare. When the counter is set to down counting, if the counter is determined as "0", the value of the <RELOADH> is loaded to the counter.

In sensor mode (phase count), the counter is captured and cleared when a rotational edge is detected on ENCLK. In phase counter mode, the counter is captured and cleared when a Z-edge is detected (ZDETECT). The captured value can be read by reading the counter register (ENxCNT).

When ENxINTCR<MCMPEN>="1", a MCMP compare match signal serves as a commutation trigger signal for the PMD circuit.

19.5.4 Interrupt Request Control Register

The counter outputs one/two interrupts among the 6 types of interrupts. A interrupt request is enabled to output with the interrupt control register (ENxINTCR) according to the interrupt event. A interrupt event can be checked with an interrupt flag (ENxINTF).

The interrupt flag register (ENxINTF) is set by the occurrence of the interrupt event, and the flag is cleared by reading the register.

Table 19-1 List of interrupt events

Interrupt event	Description	Mode	Interrupt request output enable ENxINTCR	Event flag ENxINTF	Interrupt request output
Divided pulse	When a rotational edge pulse is divided by 1 to 128 with ENxTNCR <ENDEV>, this interrupt request is sent.	Encoder mode Sensor mode(event count)	<TPLSIE>	<TPLSF>	INTENCx0
Capture	When the value of the counter is captured due to the external trigger (ENCZ input), this interrupt request is sent.	Sensor mode (phase count) Phase counter mode	<CAPIE>	<CAPF>	INTENCx0
	When the value of the counter is captured due to ENCLK, this interrupt request is sent.	Sensor mode (Timer count) Timer mode			
Detection error	When an edge detection error (PDERR) or the skip detection (SKPDT) occurs, this interrupt request is sent.	Encoder mode Sensor mode	<ERRIE>	<ERRF>	INTENCx0
A match of INT	When the value of the ENxINT register matches the counter value, this interrupt request is sent.	All modes	<CMPIE>	<INTCPF>	INTENCx1
A match of RELOAD	When the value of the ENxRELOADregister matches the count value, this interrupt request is sent.	Sensor mode (Timer count and phase count) Timer mode Phase counter mode	<RLDIE>	<RLDCPF>	INTENCx1
MCMP completion	When ENxTNCR<MCMPMD>="0", if the ENxMCMPregister matches the counter value, this interrupt request is sent. When <MCMPMD>="1", if the counter value exceeds the value of the ENxMCMP register or more, this interrupt request is sent.	Sensor mode(Timer count) Timer mode	<MCMPIE>	<MCMPF>	INTENCx1
	When the value of the ENxMCMP register matches the counter value, this interrupt request is sent.	Encoder mode Sensor mode (event count and phase count) Phase counter mode			

19.6 Example of Brushless DC Motor Control

This figure shows that a BEMF zero-cross is detected by the brushless DC motor that operates on the rectangular waveform from the PMD.

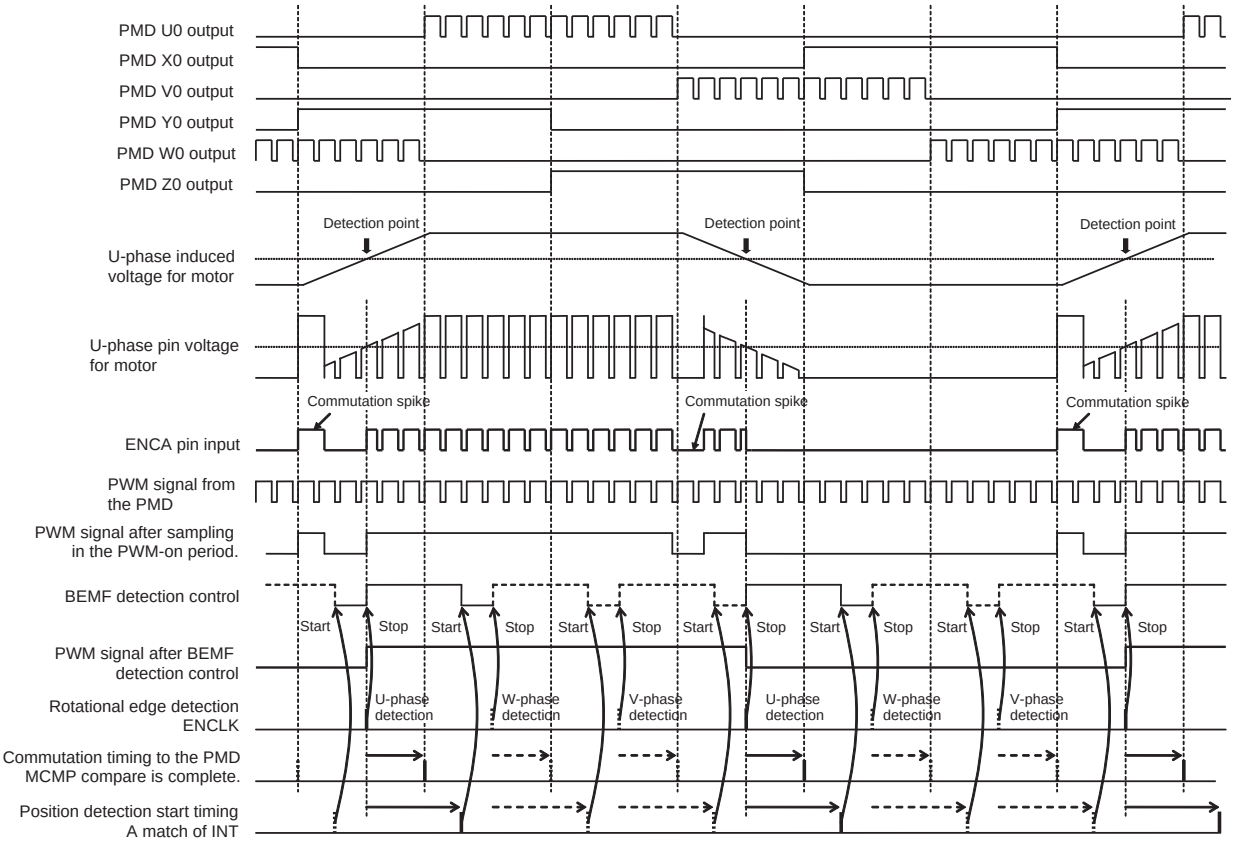


Figure 19-14 Timing waveforms of sensorless control in 120-degree energization

In sensor mode (timer count and phase count), when PWM synchronous sampling is enabled (ENxINPCR <SYNCSPLEN> ="1"), the zero-cross detection for a induced voltage (BEMF) of a brushless DC motor in the PMD that operates on rectangular waveform is supported. At this time, the input circuit removes the PWM component from the input signal using PWM synchronous sampling. In addition, the decoding circuit can avoid a commutation spike using BEMF detection control. Based on the above functions,zero-cross of BEMF can be filtered from the input signal as a rotational edge pulse (ENCLK).

BEMF detection control stops (suspends) the edge detection due to a rotational edge pulse, and it starts (restarts) the edge detection due to a match of the INT compare. It also can be stopped by the command (ENxINPCR<PDSTP> ="1"), and it can be started by the command (ENxINPCR<PDSTT> ="1").

The BEMF zero-corss function operates interlocking with the PMD circuit. Therefore, a commutation timing output should be enabled (ENxINTCR<MCMPIE> ="1").

To use the BEMF zero-cross detection, set the capture function and three compare functions of the counter as follows:

- The capture function: Capturing on a rotational edge pulse (ENCLK) (Measurement on an interval of zero-cross detection)
- MCMP compare: Commutation timing of the PMD (Example of setting: capture value × 0.5)
- INT compare: Position detection start timing (Example of setting: capture value × 0.75)
- RELOAD compare: Timeout of position detection (Example of setting: capture value × 2.0)

20. 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. Internal power supply voltage is indicated as VOUT12.

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.

20.1 Structure

Power-on-reset circuit consists of the reference voltage generation circuit and comparators.

This circuit compares a voltage divided by the ladder resistor with a reference voltage generated in the reference voltage generation circuit in the comparator.

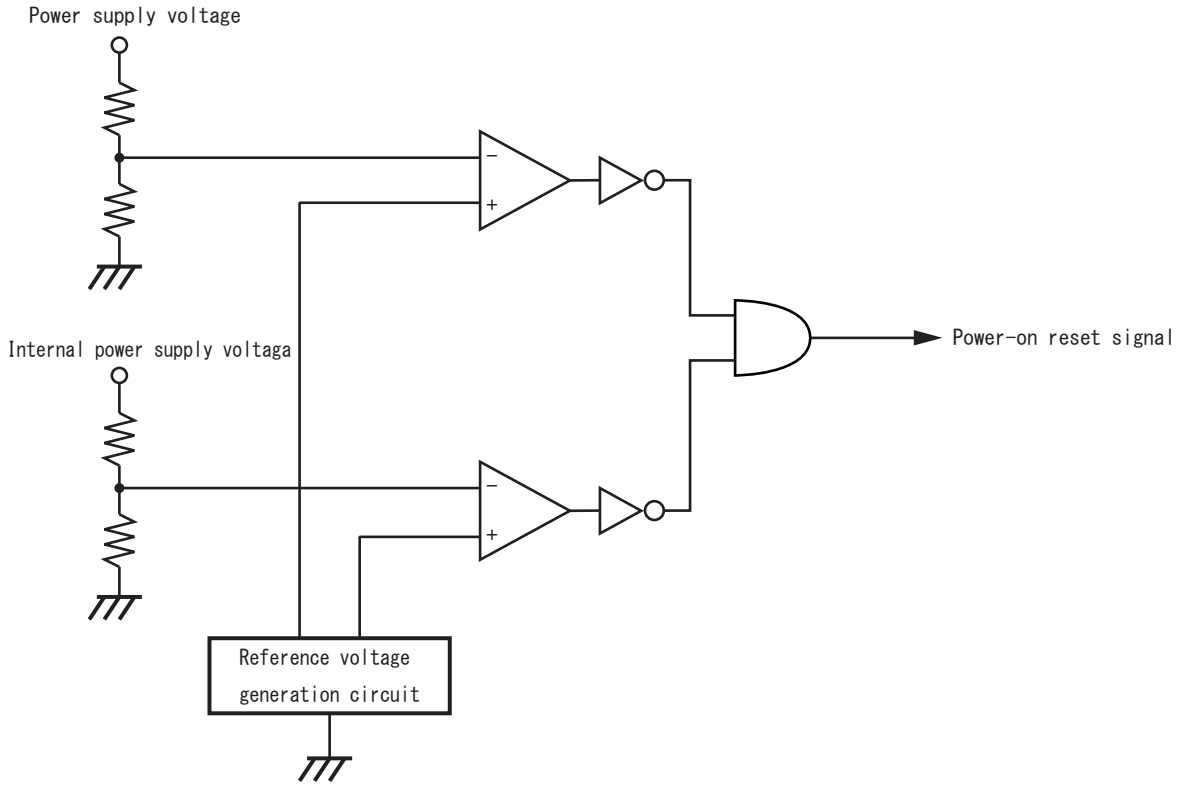


Figure 20-1 Power-on-reset circuit

20.2 Function

At power-on, a power-on reset signal is valid while power supply voltage is lower than the reset releasing voltage. Power-on reset signal is invalid at the timing when the power supply voltage is over 3.0 0.2V.

At power supply descent, a power-on reset signal is invalid while power supply voltage is upper than the reset detection voltage. Power-on reset signal is valid at the timing when the power supply voltage is less 2.8 0.2V.

During the power-on reset signal is valid, the CPU and the peripheral functions are reset.

Detail about the reset refer to the reset operation chapter.

And detail about the operation timing refer to the low voltage detection circuit (VLTD) chapter.

Note: Since the power-on reset releasing voltage and the power-on reset detection voltage relatively change, the detection voltage is never reversed.

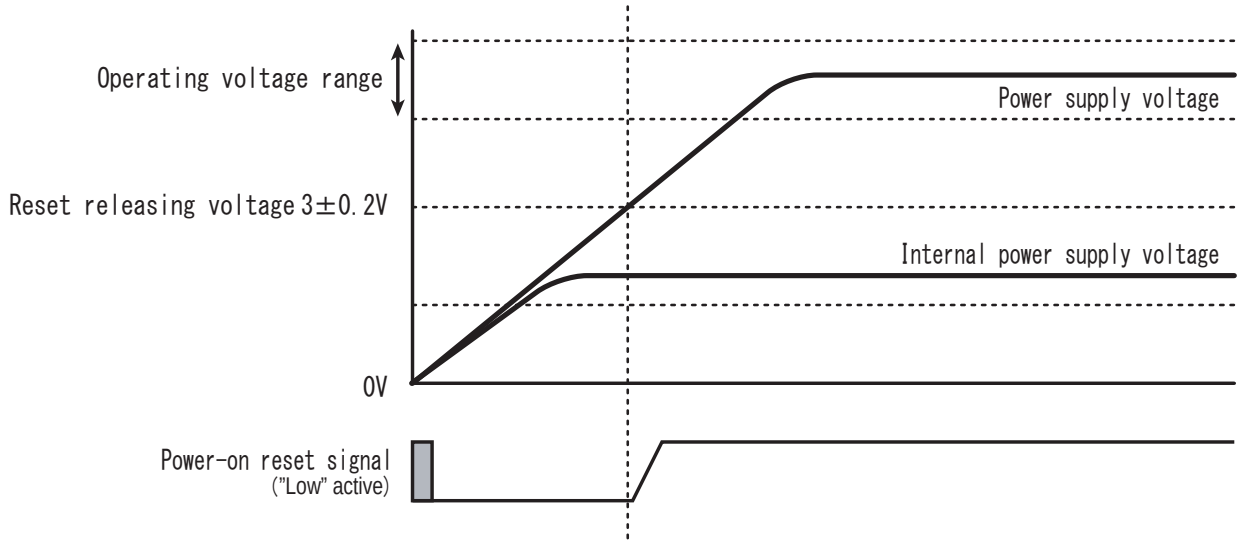


Figure 20-2 Power-on-reset operation timing at power-on

21. Low Voltage Detection Circuit (VLTD)

The low voltage detection circuit (VLTD) generates a voltage detection reset signal by detecting a decreasing power supply voltage.

Power supply voltage is indicated as DVDD5.

Note: Due to the fluctuation of power supply voltage, the voltage detection circuit may not operate properly. Users should give due consideration based on the electrical characteristic in the device designing.

21.1 Structure

The low voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

Power supply voltage 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.

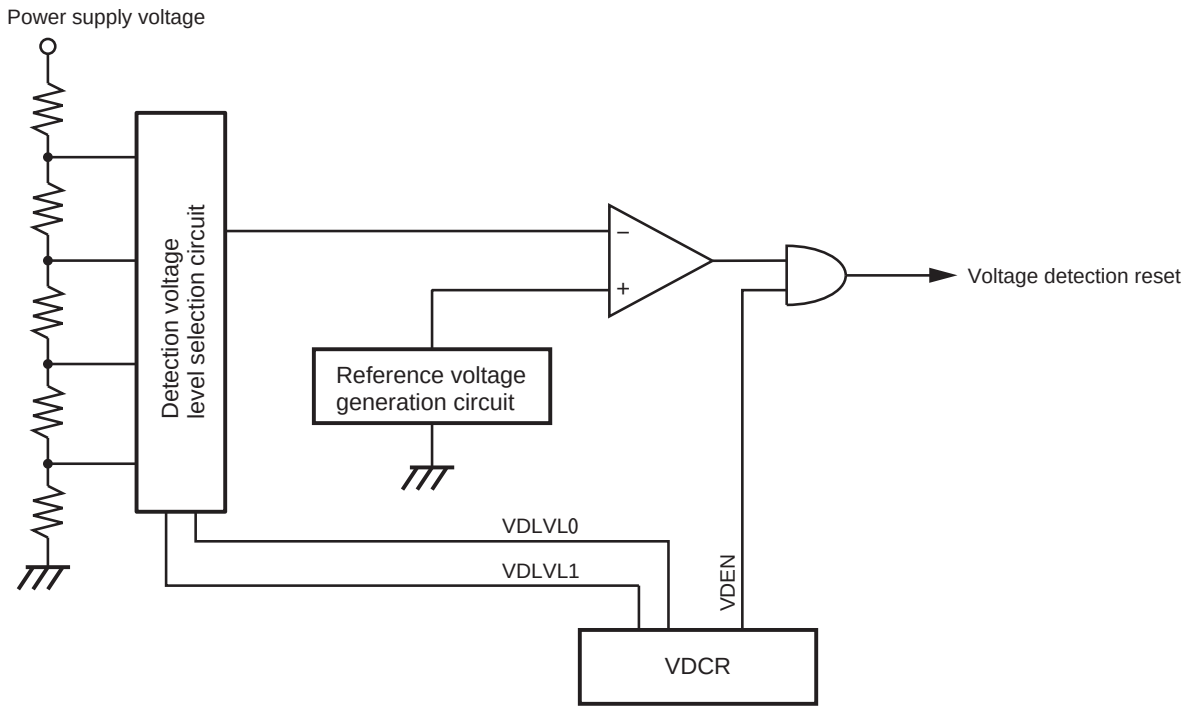


Figure 21-1 Voltage Detection Circuit

21.2 Registers

21.2.1 Register List

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+)
Voltage detection control register	VDCR	0x0000

21.2.2 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	1	1

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	Low voltage detection operation 0: Disabled 1: Enabled

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

21.3 Operation Description

21.3.1 Control

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

21.3.2 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 power supply voltage becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection reset signal is valid.

A reset is referred to "Reset Operation" chapter.

21.3.2.1 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 set to "1" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the power supply voltage is lower than the detection voltage VDCR<VDLVL[1:0]>, setting VDCR<VDEN> to "1" generates reset signal at the time.

21.3.2.2 Selecting the detection voltage level

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

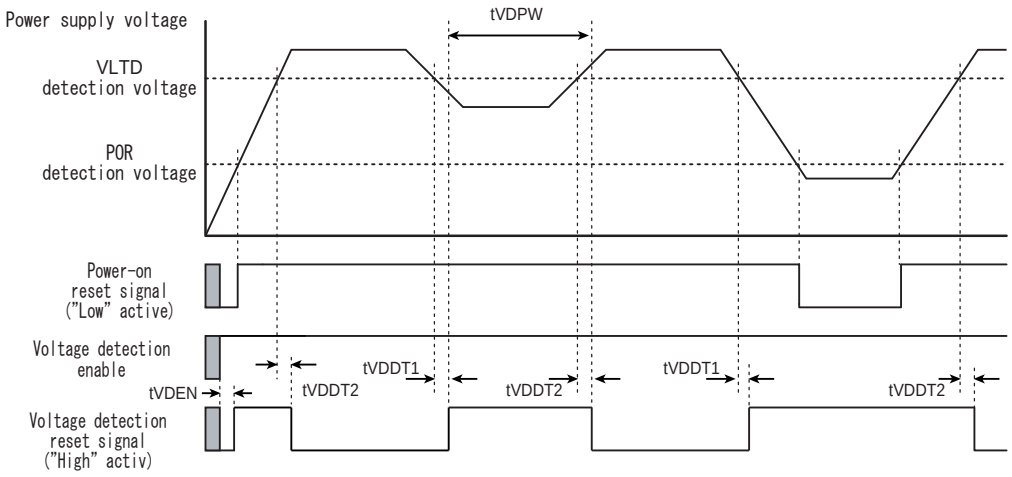


Figure 21-2 Voltage Detection Timing

Note: This timing is indicated that the power supply voltage is upper than the detection voltage VDCR<VDLVL[1:0]> after an internal reset enables the voltage detection operation.

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	-	-	

22. Oscillation Frequency Detector (OFD)

The oscillation frequency detector circuit (OFD) detects abnormal clock frequency. To use the OFD, abnormal states of clock such as a harmonic, a subharmonic or stopped state can be detected.

The OFD monitors the target clock frequency using reference frequency and generates a reset signal if abnormal state is detected. This product uses internal high-speed oscillator clock (fiosc) as a reference and the target clock are an external high-speed oscillator clock (feosc). They are selected by CGOSCCR<OSCSEL>.

Note: It is not guaranteed that OFD can detect all defects at any time, and it is not a circuit to measure error frequency.

22.1 Block diagram

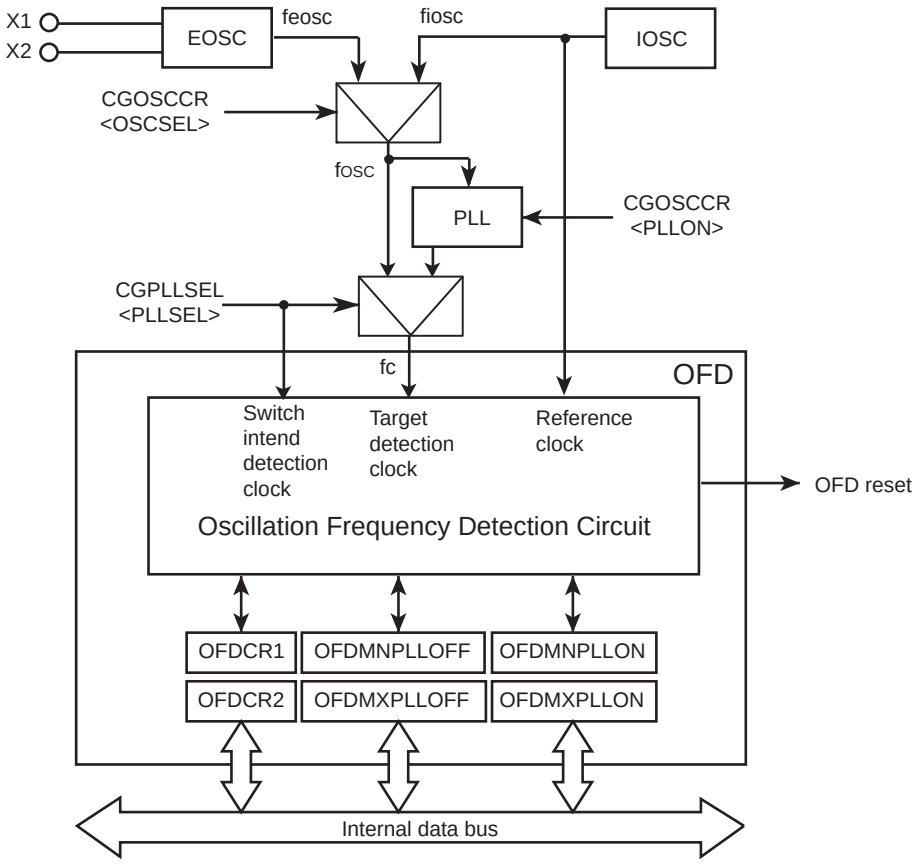


Figure 22-1 Oscillation Frequency Detector Block diagram

22.2 Registers

22.2.1 Register List

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+)
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

22.2.1.1 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 LVD reset.

22.2.1.2 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 LVD reset.

22.2.1.3 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	-	-	-	-	-	-	-	OFDMNPLL-OFF
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	0	0	0	0	0

Bit	Bit Symbol	Type	Description
31-9	−	R	Read as 0.
8-0	OFDMNPLL-OFF[8:0]	R/W	Sets internal lower detection frequency.

Note: Writing to the register of OFDMNPLLOFF is ignored while OFD circuit is operating.

Note: OFDMNPLLOFF is initialized by the $\overline{\text{RESET}}$ pin, power on reset or LVD reset.

22.2.1.4 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	0	0	0	0	0	0	0

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 ignored while OFD circuit is operating.

Note: OFDMNPLLON is initialized by the $\overline{\text{RESET}}$ pin, power on reset or LVD reset.

22.2.1.5 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	0	0	0	0	0	0

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 ignored while OFD circuit is operating.

Note: OFDMXPLLOFF is initialized by the $\overline{\text{RESET}}$ pin, power on reset or LVD reset.

22.2.1.6 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	-	-	-	-	-	-	-	OFDMXPL-LON
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	OFDMXPLLON							
After reset	0	0	0	0	0	0	0	0

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 ignored while OFD circuit is operating.

Note: OFDMXPLLON is initialized by the $\overline{\text{RESET}}$ pin, power on reset or LVD reset.

22.3 Operational Description

22.3.1 Setting

Registers of the oscillation frequency detector circuit (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. These registers are automatically switched over by the setting of CGPLLSEL<PLLSEL>. If OFD reset is generated with PLL-ON (CGPLLSEL<PLLSEL> = "1"), the detection frequency setting registers (OFDMNPLLON/OFDMXPLLON) are automatically switched over to OFDMNPLLOFF/OFDMXPLLOFF with PLL-OFF (CGPLLSEL<PLLSEL> = "0").

Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection.

To protect against the mistaken writing, should be written "0x06" to OFDCR1. And the register should be modified when OFD is stopped.

When STOP mode is executed with OFDCR2=0xE4, OFD is automatically disabled. After releasing STOP mode and warming up period, OFD is enabled.

Note: When the PLL is controlled (enabled or disabled) by the CGPLLSEL register or when the system clock is changed (feosc or fiosc) by the <OSCSEL> of CGOSCCR register, the OFD must be disabled beforehand.

22.3.2 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.

Table 22-1 Availability of oscillation frequency detector

Operating Mode	Oscillation Frequency Detection (OFDCR2=0xE4)	All I/Os condition after Oscillation Frequency Detection RESET (Except power supply, $\overline{\text{RESET}}$, MODE 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	Available	High impedance
Watchdog timer reset SYSRESETREQ reset	Available	High impedance
RESET by external reset power on reset LVD reset	Disable	-

22.3.3 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.

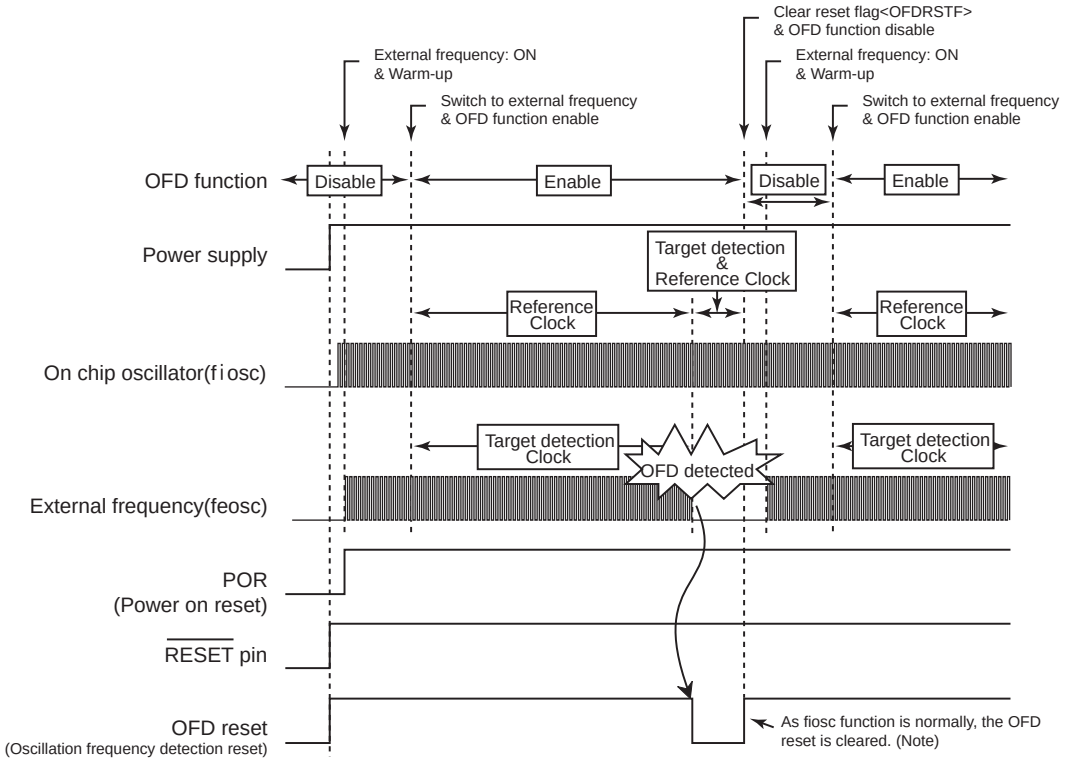
When generating reset is enabled, OFD generates reset if the target clock frequency exceeds the frequency limit set by OFDMNPLLON/OFDMXPLLON and OFDMNPLLOFF/OFDMXPLLOFF. From detection of abnormal frequency to reset generation, time length as one cycle of detecting clock is needed. The reset generated by OFD does not make itself and OFD continues detected operation.

If OFD generates reset when the target clock(fc) is feosc in PLL ON, fosc is initialized to fiosc and fc changes to fiosc automatically in PLL OFF, so the target clock becomes same as the reference clock. The frequency range is instated within the set value and OFD reset is release.

When the internal high-speed oscillator clock (fiosc) is selected as a system clock(fc), OFD does not detect.

Note 1: There are several factors of reset. Clock generator register CGRSTFLG can confirm the factors. For details of CGRSTFLG see chapter "exception."

Note 2: 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 fiosc 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 feosc and the reset generated by OFD, the reset might be generated continuously due to as abnormal detection until OFD detects correctly.

Figure 22-2 Example of oscillation frequency detection operation

22.3.4 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 FDMNPLLON/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 ±5% (undetecting range) and the reference clock error is ± 5%. In this example, OFDMXPLLOFF is rounded up and OFDMNPLLOFF is rounded down.(From "a" to "h" corresponds to "Figure 22-3 Example of detection frequency range (in case of 10MHz)")

target clock	10MHz ± 5%	Max 10.5MHz	-------- c
		Min 9.5MHz	-------- b
reference clock	9.7MHz ± 5%	Max 10.185MHz	-------- f
		Min 9.215MHz	-------- e

OFDMXPLLOFF = c ÷ e × 32 = 36.46... = 37 (Rounding up to nearest decimal) = 0x25

OFDMNPLLOFF = b ÷ f × 32 = 29.85... = 29 (Rounding down to nearest decimal) = 0x1D

At this time, the detecting range is calculated shown below.

a = e × OFDMNPLLOFF ÷ 32 = 8.35

d = f × OFDMXPLLOFF ÷ 32 = 11.78

And the undetecting range is calculated shown below.

g = e × OFDMXPLLOFF ÷ 32 = 10.65

h = f × OFDMNPLLOFF ÷ 32 = 9.23

Setting "0x25" to the register OFDMXPLLOFF and "0x1D" to the register OFDMNPLLOFF, when the target clock of higher than 11.78MHz or lower than 8.35MHz is detected, the oscillation frequency detector outputs a reset signal. And when the target clock of higher than 9.23MHz and lower than 10.65MHz is detected, the oscillation frequency detector does not output a reset signal.

Figure 22-3 shows the detection or undetectable and detectable frequency range.

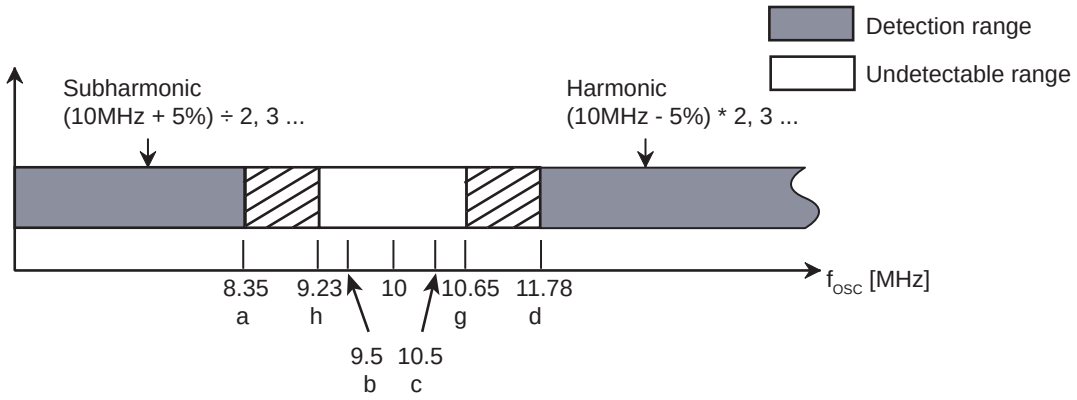


Figure 22-3 Example of detection frequency range (in case of 10MHz)

22.3.5 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.

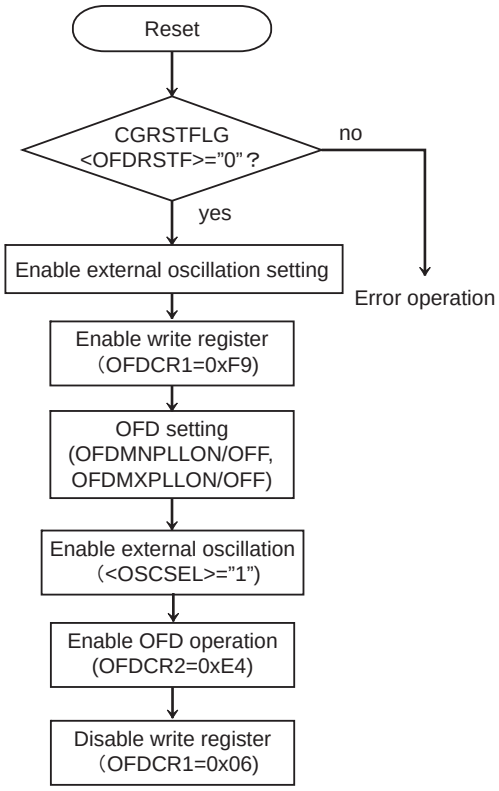


Figure 22-4 Example of operational procedure

23. Watchdog Timer (WDT)

The watchdog timer (WDT) is for detecting malfunctions (runaway) 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: TMPM475FDFG/FZFG/FYFG does not have the watchdog timer out pin ($\overline{\text{WDTOUT}}$).

23.1 Configuration

Figure 23-1 shows the block diagram of the watchdog timer.

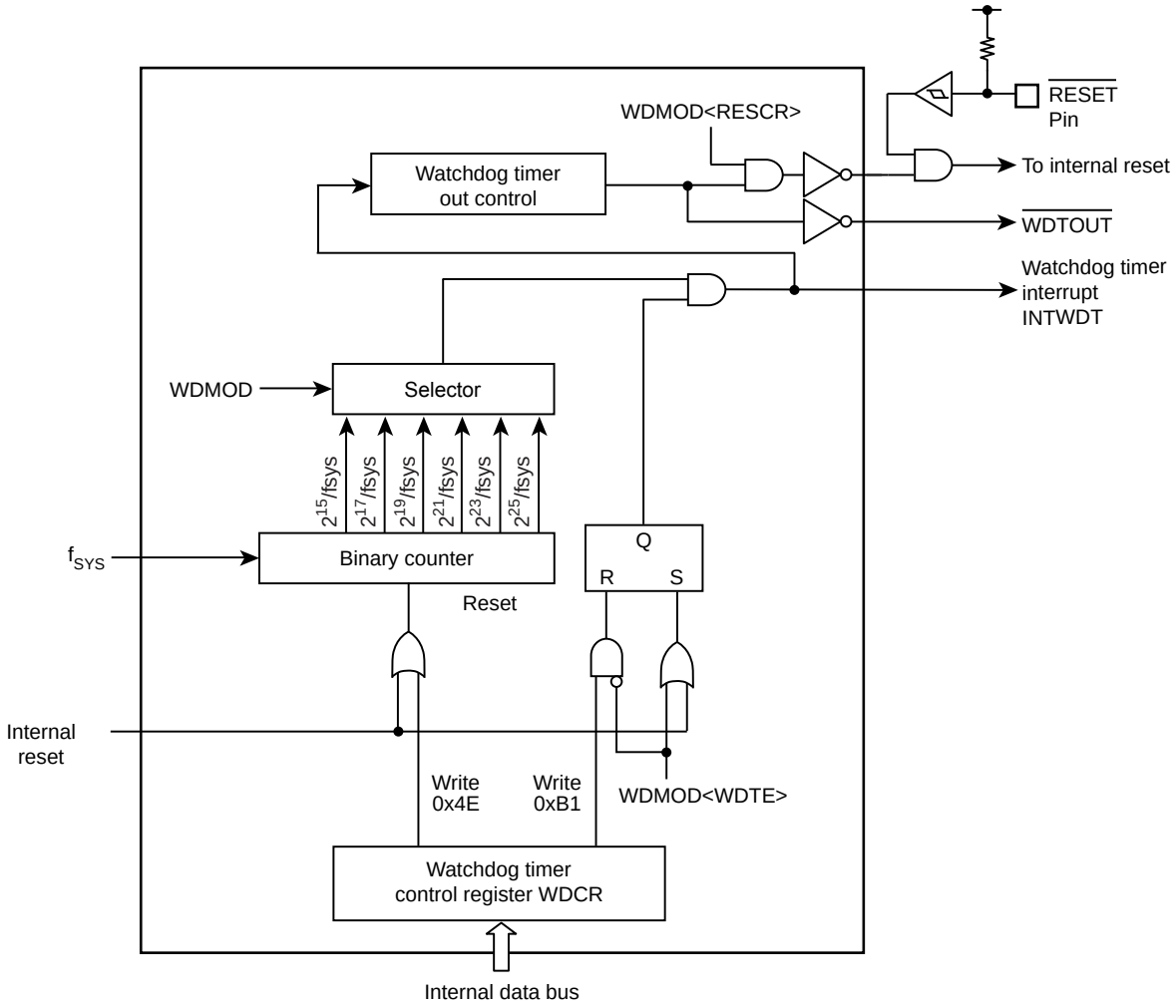


Figure 23-1 Block Diagram of the watchdog Timer

23.2 Register

23.2.1 Register 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+)
Watchdog Timer Mode Register	WDMOD	0x0000
Watchdog Timer Control Register	WDCR	0x0004

23.2.2 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 To disable the watchdog timer to protect from the error writing by the malfunction, first <WDTE> is set to "0", and then the disable code (0xB1) must be written to WDCR. To change the status of the watchdog timer from "disable" to "enable", set <WDTE> to "1".
6-4	WDTP[2:0]	R/W	Selects WDT detection time 000: $2^{15}/f_{SYS}$ 100: $2^{23}/f_{SYS}$ 001: $2^{17}/f_{SYS}$ 101: $2^{25}/f_{SYS}$ 010: $2^{19}/f_{SYS}$ 110: Reserved 011: $2^{21}/f_{SYS}$ 111: Reserved
3	−	R	Read as "0"
2	I2WDT	R/W	Operation in IDLE mode 0: Stop 1: Operate
1	RESCR	R/W	Operation after detecting malfunction 0: INTWDT interrupt request is generated. Note) 1: Reset
0	−	R/W	Write "0".

Note: INTWDT interrupt is a factor of the non-mask interrupt.

23.2.3 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

23.3 Description of Operation

23.3.1 Basic Operation

The watchdog timer is consists of the binary counter that works 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) is generated, and the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs "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 is generated. If the binary counter is not cleared, the non-maskable interrupt is generated by INTWDT. Thus CPU detect malfunction (runaway), 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: TMPM475FDFG/FZFG/FYFG does not have a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

23.3.2 Operation Mode and Status

The watchdog timer begins operation immediately after a reset is released. If not using the watchdog timer, it should be disabled.

The watchdog timer cannot be used at the high-speed frequency clock is stopped. Before transition to below operation modes, the watchdog timer should be disabled.

In IDLE mode, its operation depend on WDMOD<I2WDT> setting.

- STOP mode

Also, the binary counter is automatically stopped during debug mode.

23.3.3 Operation when malfunction (runaway) is detected.

23.3.3.1 INTWDT interrupt generation

Figure 23-2 shows the case that INTWDT interrupt is generated (WDMOD<RESCR>="0").

When an overflow of the binary counter occurs, INTWDT interrupt is generated. It is a factor of non-maskable interrupt (NMI). Thus CPU detects non-maskable interrupt and perform the countermeasure program.

When INTWDT interrupt is generated, 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.

Note: TMPM475FDFG/FZFG/FYFG does not have a watchdog timer out pin ($\overline{\text{WDTOUT}}$).

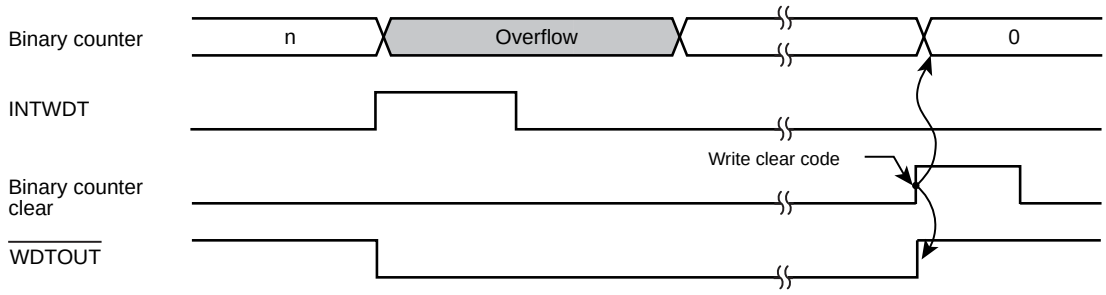


Figure 23-2 INTWDT interrupt generation

23.3.3.2 Internal Reset Generation

Figure 23-3 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.

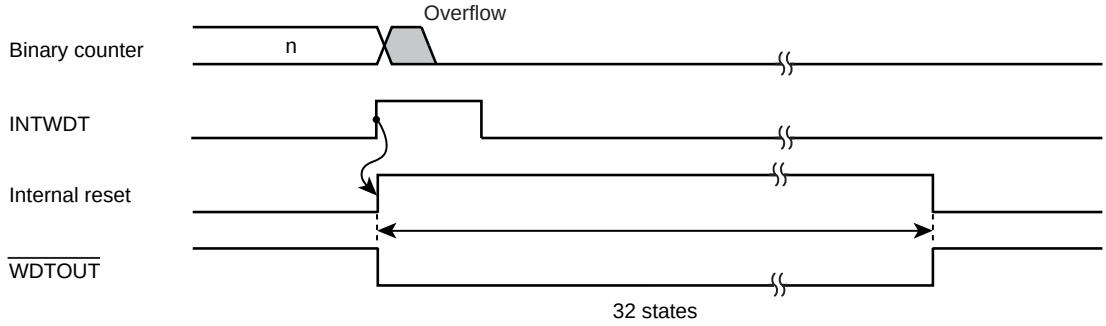


Figure 23-3 Internal reset generation

23.4 Control of the watchdog timer

23.4.1 Disable control

By writing the disable code (0xB1) to WDCR after setting WDMOD<WDTE> to "0", the watchdog timer can be disabled and the binary counter can be cleared.

23.4.2 Enable control

Set WDMOD<WDTE> to "1".

23.4.3 Watchdog timer clearing control

Writing the clear code (0x4E) to WDCR clears the binary counter and it restarts counting.

23.4.4 Detection time of watchdog timer

Set WDMOD<WDTP[2:0]> depend on the detection time.

For example, in the case that $2^{21}/f_{SYS}$ is used, set "011" to WDMOD<WDTP[2:0]>.

24. Flash Memory (FLASH)

This section describes the hardware configuration and operation of Flash memory. In this section, "1-word" means 32 bits.

24.1 Features

24.1.1 Memory Size and Configuration

Table 24-1 and Table 24-2 and show a built-in memory size and configuration of TMPM475FDFG/FZFG/FYFG.

Table 24-1 Memory size and configuration

Product	Memory size (KB)	Area information		Block information		Page information		Program time (s) (Note)	Erase time (ms) (Note)			
		Size (KB)	# of memory	Size (KB)	# of memory	Size (Byte)	# of memory		1 page	1 block	1 area	Chip
TMPM475FDFG	512	512	1	32	16	4096	128	8.0	115	920	115	115
TMPM475FZFG	384	384	1		12		96	6.0				
TMPM475FYFG	256	256	1		8		64	4.0				

Note: Above time is assumed as an initial value of each register after reset. Data transfer time is not included.
Flash programming time per chip varied depending on users' programming method.

Table 24-2 Block Configuration(1024KB)

Area No.	Block-No.	Address (Single chip mode)	Address (Single boot mode) (Single chip mode(mirror))	Size (KByte)	Number of pages	Usable areas		
						TMPM475FDFG	TMPM475FZFG	TMPM475FYFG
0	0	0x0000_0000 to 0x0000_7FFF	0x5E00_0000 to 0x5E00_7FFF	32	8	*	*	*
	1	0x0000_8000 to 0x0000_FFFF	0x5E00_8000 to 0x5E00_FFFF	32	8	*	*	*
	2	0x0001_0000 to 0x0001_7FFF	0x5E01_0000 to 0x5E01_7FFF	32	8	*	*	*
	3	0x0001_8000 to 0x0001_FFFF	0x5E01_8000 to 0x5E01_FFFF	32	8	*	*	*
	4	0x0002_0000 to 0x0002_7FFF	0x5E02_0000 to 0x5E02_7FFF	32	8	*	*	*
	5	0x0002_8000 to 0x0002_FFFF	0x5E02_8000 to 0x5E02_FFFF	32	8	*	*	*
	6	0x0003_0000 to 0x0003_7FFF	0x5E03_0000 to 0x5E03_7FFF	32	8	*	*	*
	7	0x0003_8000 to 0x0003_FFFF	0x5E03_8000 to 0x5E03_FFFF	32	8	*	*	*
	8	0x0004_0000 to 0x0004_7FFF	0x5E04_0000 to 0x5E04_7FFF	32	8	*	*	
	9	0x0004_8000 to 0x0004_FFFF	0x5E04_8000 to 0x5E04_FFFF	32	8	*	*	
	10	0x0005_0000 to 0x0005_7FFF	0x5E05_0000 to 0x5E05_7FFF	32	8	*	*	
	11	0x0005_8000 to 0x0005_FFFF	0x5E05_8000 to 0x5E05_FFFF	32	8	*	*	
	12	0x0006_0000 to 0x0006_7FFF	0x5E06_0000 to 0x5E06_7FFF	32	8	*		
	13	0x0006_8000 to 0x0006_FFFF	0x5E06_8000 to 0x5E06_FFFF	32	8	*		
	14	0x0007_0000 to 0x0007_7FFF	0x5E07_0000 to 0x5E07_7FFF	32	8	*		
	15	0x0007_8000 to 0x0007_FFFF	0x5E07_8000 to 0x5E07_FFFF	32	8	*		

Flash memory configuration units are described as "block" and "page".

- Page

  Used in erase function and protect function.

  One page is fixed to 4096 bytes.
- Block

  Used in erase function and protect function.

  One block is fixed to 32K bytes.
- Area

  Used in erase function.

  There are 512K byte areas.

Write operation is performed in the units of 16 bytes (4 bytes x 4 times). The write time per 16 bytes is 163µs (Typ.).

Erase is performed per block or performed on entire flash memory. Erase time varies on commands. If auto block command is used, the erase time will be 920 ms. per block (Typ.). If the auto chip erase command is used to erase entire area, the time will be 345 ms. (Typ.). If the other commands is used, the time will be 115 ms (Typ.).

In addition, the protect function is set by a page from page 0 to 7. The other blocks are set by a block. When the protection function is erased, the entire Flash memory is erased once. For detail of the protect function, refer to "24.1.5 Protect/Security Function".

24.1.2 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	<Addition> Auto area erase, auto page erase, auto memory swap <Modified> Block write/erase protect (only software protection is supported) <Deleted> Erase resume - suspend function

24.1.3 Operation Mode

Do not power off or reset the MCU during the flash memory write/erase operation.

24.1.3.1 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 24-1 shows the mode transition.

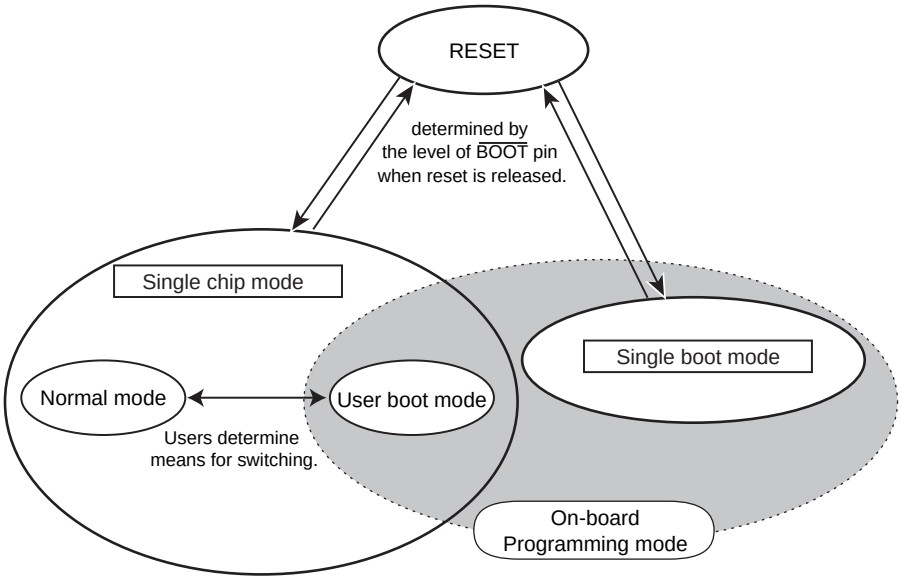


Figure 24-1 Mode transition

(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 3 sub-modes in below.

- Normal mode

  The mode where user application program is executed.

- User boot mode

User boot mode is the mode in which Flash memory externally located in the Flash memory can be re-programmed.

For details of Flash reprogramming, refer to "24.4 Reprogramming in the User Boot Mode". 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.

Switching each mode is determined freely by users. For example, when PA0 of port A is set to "1", the mode is normal mode; when PA0 of port A is set to "0", the mode is user boot mode. Users should be provide the routine in the part of application program to switch the mode.

(2) Single boot mode

Single boot mode is the mode in which Flash memory externally located in the Flash memory can boot-up from the built-in BOOT ROM (Mask ROM) after reset.

Flash memory can be reprogrammed by Flash memory reprogramming program located in outside of Flash memory such as single internal memory. For details about reprogramming Flash memory, refer to "24.3 How to Reprogram Flash in Single Boot Mode".

The BOOT ROM contains the algorithm that can rewrite Flash memory via serial port of this device on the users' set. With connecting the serial port to external host, data transfer is performed in above-mentioned protocol and re-programmed 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 users' set. These two modes are called "on-board programming mode".

24.1.3.2 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.

For details of reset operation, refer to the chapter on "Reset Operation".

(1) Mode determination at warm reset

Table 24-3 Operation mode setting at warm reset

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

(2) Mode determination at cold reset

Table 24-4 Operation mode setting at cold reset (not using $\overline{\text{RESET}}$ pin)

Operation mode	POR extension signal	Pin
		$\overline{\text{BOOT}}$
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

- Mode determination by the power-on reset circuit (the $\overline{\text{RESET}}$ pin is not used).

  Maintain the $\overline{\text{BOOT}}$ pin setting until the POR extension signal from the power-on counter is changed to "1" from "0".

- Mode determination by the $\overline{\text{RESET}}$ pin

  When the POR extension signal is "1", reset operation using the $\overline{\text{RESET}}$ pin can be used. Therefore, execute the same setting described in Table 24-3.

  When the POR extension signal is "0", execute the same setting described in Table 24-4. In this case, reset operation by the POR extension signal is prior to those by the $\overline{\text{RESET}}$ pin.

24.1.4 Memory Map

Figure 24-2 and Figure 24-3 show comparisons 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 0x5E00_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.

Product	Flash size	RAM size	Flash address	RAM address
TMPM475FDFG	512KB	34KB	0x0000_0000 to 0x0007_FFFF (Single chip mode) 0x5E00_0000 to 0x5E07_FFFF (Single chip mode(mirror)) 0x5E00_0000 to 0x5E07_FFFF (Single boot mode)	0x2000_0000 ~ 0x2000_87FF
TMPM475FZFG	384KB	34KB	0x0000_0000 to 0x0005_FFFF (Single chip mode) 0x5E00_0000 to 0x5E05_FFFF (Single chip mode(mirror)) 0x5E00_0000 to 0x5E05_FFFF (Single boot mode)	0x2000_0000 ~ 0x2000_87FF
TMPM475FYFG	256KB	18KB	0x0000_0000 to 0x0003_FFFF (Single chip mode) 0x5E00_0000 to 0x5E03_FFFF (Single chip mode(mirror)) 0x5E00_0000 to 0x5E03_FFFF (Single boot mode)	0x2000_0000 ~ 0x2000_3FFF 0x2000_8000 ~ 0x2000_87FF

Note: In TMPM475FZFG and TMPM475FYFG, there is a common memory area for ID and password (0x5E07_FFF0 to 0x5E07_FFFF).

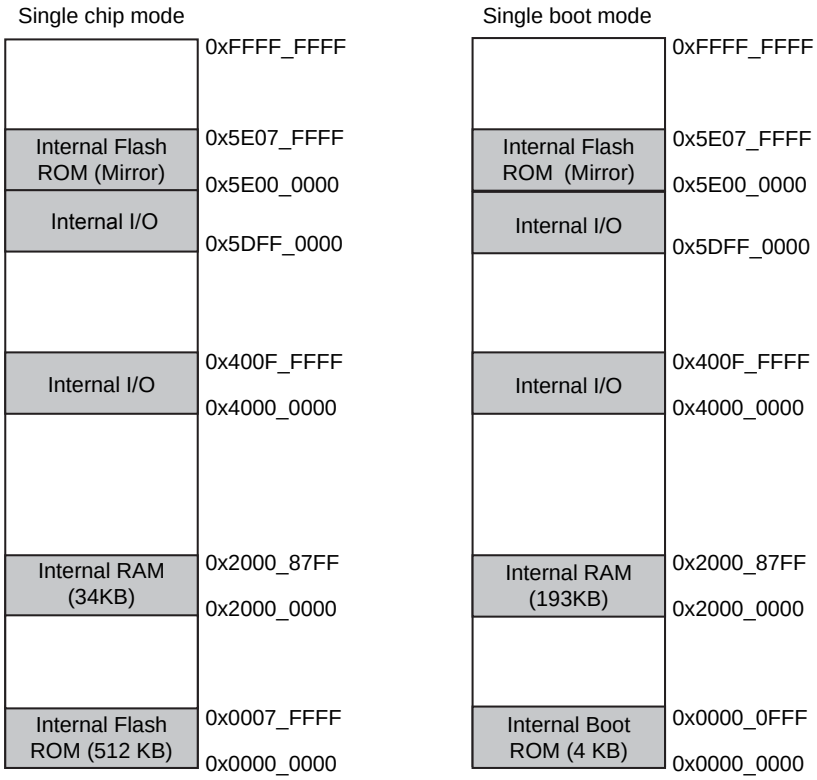


Figure 24-2 Comparison of memory map (TMPM475FDFG)

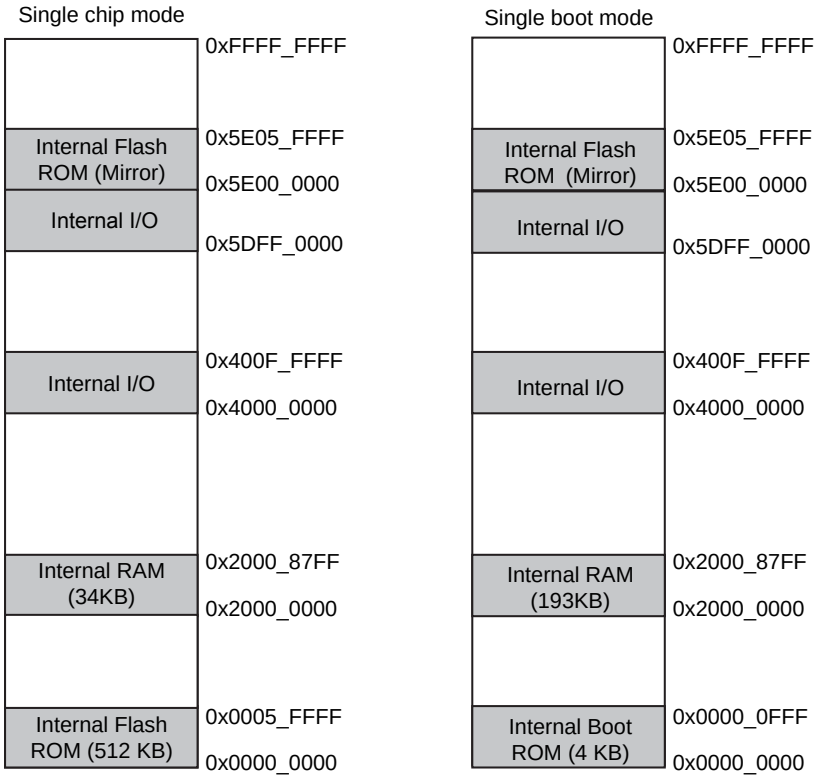


Figure 24-3 Comparison of memory map (TMPM475FZFG)

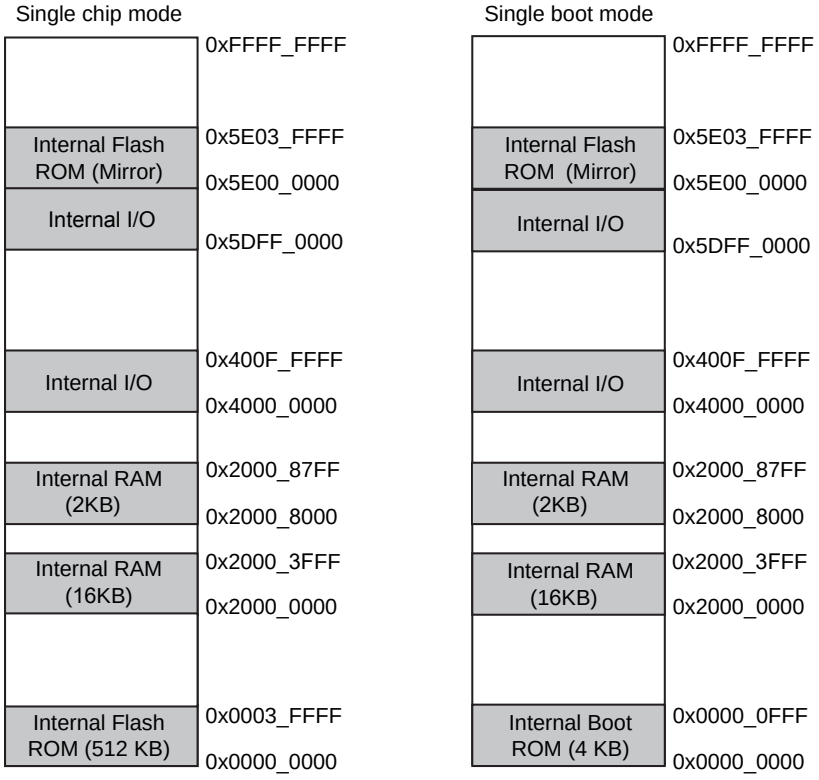


Figure 24-4 Comparison of memory map (TMPM475FYFG)

24.1.5 Protect/Security Function

This device has the protect and security functions for Flash memory.

1. Protect function

   The write/erase operation can be prohibited per block.

2. Security function

   The read operation from a flash writer can be prohibited.

   Usage restrictions on debug functions

24.1.5.1 Protect Function

Block 0 can use the protect function by page. From Block1 through the last block, these blocks can use the protect function by 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 cancelled. The protect bit can be monitored in each FCPSR register.

For detail of programming/erasing of protect bits, refer to Chapter "24.2.6 Command Description".

24.1.5.2 Security Function

Table 24-5 shows operations when the security function is enabled.

Table 24-5 Operations when the security function is enabled.

Item	Description
Read flash memory	CPU can read flash memory.
Debug port	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 (all bits in each FCPSR register) are set to "1".

FCSECBIT<SECBIT> is set to "1" by the cold reset. Rewriting of FCSECBIT<SECBIT> is described in below.

Note: Use a 32-bit transfer instruction to 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.

Note: When FCSECBIT<SECBIT> is "0", protect bits are masked in "0" and the write/erase operation is enabled. (The security and protect function are disabled.) However, original protect bits do not change.

Table 24-6 <SECBIT> and protect bits

	<SECBIT>=1	<SECBIT>=0
Protect bits <All>=1	The security function is enabled.	Serial wire or trace communication is enabled. The write/erase operation is enabled.
Protect bits <any>=1	Serial wire or trace communication is enabled. The write/erase operation is prohibited per block.	Serial wire or trace communication is enabled. The write/erase operation is enabled.
Protect bits <All>=0	Serial wire or trace communication is enabled. The write/erase operation is enabled.	Serial wire or trace communication is enabled. The write/erase operation is enabled.

24.1.6 Memory Swap Function

24.1.6.1 Outline

If the power is off in the middle of Flash memory reprogramming, programming may not be executed. Suppose you came upon a situation in which the power becomes OFF after a program is erased. In this case, you cannot finish the programming. To avoid such case, use this memory swap function to save your program.

24.1.6.2 Operation Description

A swap region is determined by the region starting from address 0 and the next region. A swap size is determined by FCSWPSR<SIZE>. To change this size, set "1" to FCSWPSR<SIZE> bit using automatic memory swap command.

In order to perform memory swap, set "1" to FCSWPSR[0] using automatic memory swap command. To release the swap condition, set "1" to FCSWPSR[1] using automatic memory swap command. Swap condition can be checked by FCSWPSR<SWP>.

For details of automatic memory swap command, "24.2.6 Command Description".

24.1.6.3 Usage of Memory Swap

This section describes the basic flow of memory swap procedure. For an example of memory swap, refer to "24.5 How to Reprogram Flash using User Boot Mode".

1. Release the security function if the security function is enabled.

   For details of security releasing, refer to "24.1.5.2 Security Function".

   If the security function is not released, Flash memory is erased according to the command execution.

2. If the security function is enabled, erase the protect bit.

   For details of erasing protect bit, refer to "24.1.5.1 Protect Function".

   If the protect bit is not erased, command execution is not performed according to the procedure.

3. Confirm if the next region next to the region starting from address 0 is blank. (Hereafter, a region starting from address 0 is called Page0; the next region is called Page1) If this region is not blank, erase the data.

   Page0:Old original data

   Page1:Blank

4. Write the original data starting from address 0 to the next region (Both regions become the same data.)

   Page0:Old original data

   Page1:Copy data (Old original data)

5. Perform memory swap

   Page0:Copy data (Old original data)

   Page1:Old original data

6. Erase old original data to be blank

   Page0:Copy data (Old original data)

   Page1:Blank

7. Write new data to the blank region

   Page0:Copy data (Old original data)

   Page1:New original data

8. Release swap conditions

   Page0:New original data

   Page1:Copy data (old original data)

9. Execute automatic protect bit erase command

10. Options if required
    - Erase copy data (old original data).
    - Reprogramming Flash memory data except swap regions
    - Validate the protect function
    - Validate the security function

Procedure		3	4	5	6	7	8
On-chip RAM		Erase routine	Programming routine	Swap routine	Erase routine	Programming routine	Swap routine
Flash memory	Page0	Old original	Old original	Copy of old original	Copy of old original	Copy of old original	New original
	Page1	Blank	Copy of old original	Old original	Blank	New original	Copy of old original

Erase routine: A program is to erase Flash memory.

Programming routine: A program is to program Flash memory.

Swap routine: A program is to swap Flash memory.

24.1.7 Register

24.1.7.1 Register List

The following table shows control registers and addresses.

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

Peripheral function : FC

Register name		Address (Base+)
Security bit register	FCSECBIT	0x0010
Protect status register 0	FCPSR0	0x0020
Status register	FCSR	0x0100
Swap status register	FCSWPSR	0x0104
Area selection register	FCAREASEL	0x0140
Control register	FCCR	0x0148
Status clear register	FCSTSCLR	0x014C
WCLK setting register	FCWCLKCR	0x0150
Count setting register for program	FCPROGCR	0x0154
Count setting register for erase	FCERASECR	0x0158

24.1.7.2 FCSECBIT (Security Bit Register)

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

Bit	Bit Symbol	Type	Function
31-1	−	R	Read as "0".
0	SECBIT	R/W	Security bit 0: Security function setting is disabled. 1: Security function setting is enabled.

Note: This register is initialized by cold reset.

24.1.7.3 FCPSR0 (Protect Status Register 0)

	31	30	29	28	27	26	25	24
bit symbol	BLK15	BLK14	BLK13	BLK12	BLK11	BLK10	BLK9	BLK8
After reset	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)	(Note1)
	23	22	21	20	19	18	17	16
bit symbol	BLK7	BLK6	BLK5	BLK4	BLK3	BLK2	BLK1	-
After reset	(Note1)	(Note1)	(Note1)	(Note1)	(Note)	(Note1)	(Note1)	0
	15	14	13	12	11	10	9	8
bit symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
After reset	(Note1)	(Note1)	(Note1)	(Note1)	(Note)	(Note1)	(Note1)	(Note1)
	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
31-17	BLK15 to BLK1	R	Protection status of Block1 to 15 0: No protection 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.
16	−	R	Read as "0".
15-8	PG7 to PG0	R	Protection status of Page0 to 7 0: No protection 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.
7-1	−	R	Read as "0".
0	RDY_BSY	R	Ready/Busy flag when automatic program or automatic chip erase command is executing. (Note2) 0: In automatic operation 1:Automatic 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.

Note 1: Depend on the Protection status

Note 2: Make sure that Flash memory is ready before commands are issued. .

24.1.7.4 FCSR (Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	WEABORT
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	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
31-25	−	R	Read as "0".
24	WEABORT	R	Once the auto operation is aborted, "1" is set to this bit. For details, refer to "24.2.4 Abortion of Automatic Operation".
23-0	−	R	Read as "0x000001".

24.1.7.5 FCSWPSR (Swap 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	-	-	-	-	-	SIZE		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	FLG						SWP	
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-11	−	R	Read as "0".
10-8	SIZE[2:0]	R	Swap size 000: 4K bytes 001: 8K bytes 010: 16K bytes 011: 32K bytes Other than the above: Setting is prohibited.
7-2	FLG	R	Use as a flag bit for software management(refer to the "24.5 How to Reprogram Flash using User Boot Mode" for an example.)
1-0	SWP[1:0]	R	Swap state 11: Release the swap 10: Setting is prohibited. 01: In swapping 00: Release the swap (Initial state)

Note: This register is initialized by auto protect bit erase command.

24.1.7.6 FCAREASEL (Area Selection 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	-	-	-	-	-	AREA2		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	AREA1			-	AREA0		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-0	AREA0	R/W	Specifies an "area" in the Flash memory that is targeted by Flash memory operation command. 111: Selects area 0. Other than the above: non-selective area 0.

Note 1: If area selection bit (<AREAn>) is written, wait for written data to become read by polling.

Note 2: If Flash memory operation command is executed when area selection bit is other than "111" (0x7), the command execution is cancelled.

Note 3: When auto chip erase command is executed, set "111" (0x7) to all area selection bit.

24.1.7.7 FCCR (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	-	-	-	-	-	WEABORT		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-0	WEABORT	R/W	Abort of auto operation command 111: Abort Read as the setting value. For details, refer to "24.2.4 Abortion of Automatic Operation".

24.1.7.8 FCSTSCLR (Status 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	-	-	-	-	-	WEABORT		
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as "0".
2-0	WEABORT	R/W	Clears FCSR<WEABORT> to "0". 111: Clear Read as the setting value. For details, refer to "24.2.4 Abortion of Automatic Operation".

24.1.7.9 FCWCLKCR (WCLK 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	-	-	-	DIV				
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-5	−	R	Read as "0".
4-0	DIV	R/W	Frequency division ratio to change the clock (WCLK: fsys/(DIV+1)) in automatic operation to 8 to 12MHz 00000: Divide-by-1 00001: Divide-by-2 : 11110: Divide-by-31 11111: Divide-by-32

Note 1: Before setting <DIV>, make sure to complete the gear change process.

Note 2: When Flash operation is performed with changing gear, re-set WCLK to be within 8 to 12MHz according to the operation frequency (fsys). Table 24-6 shows an example of general operation frequency (fsys) and <DIV>.

Table 24-6 General clock in automatic operation corresponding to operation frequency (fsys) and <DIV>

	fsys	10MHz	20MHz	25MHz	30MHz	40MHz	80MHz	100MHz	120MHz
<DIV>	Divide ratio	Clock in automatic operation (MHz)							
00000	Divide-by-1	10	-	-	-	-	-	-	-
00001	Divide-by-2	-	10	-	-	-	-	-	-
00010	Divide-by-3	-	-	8.3	10	-	-	-	-
00011	Divide-by-4	-	-	-	-	10	-	-	-
00111	Divide-by-8	-	-	-	-	-	10	-	-
01001	Divide-by-10	-	-	-	-	-	8	10	12
01011	Divide-by-12	-	-	-	-	-	-	8.3	10
01110	Divide-by-15	-	-	-	-	-	-	-	8

-: Setting is prohibited.

24.1.7.10 FCPROGCR (Count Setting Register for Program)

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

Bit	Bit Symbol	Type	Function
31-2	−	R	Read as "0".
1-0	CNT	R/W	Set the number of counts that makes a programming time (CNT/WCLK) by automatic program execution command be within the range of 20 to 40 μs. 00: Number of counts 250 01: Number of counts 300 Other than the above: Number of counts 350

Note : When WCLKCR<DIV> is re-set, re-set a programming time by automatic program execution command within the range of 20 to 40μs if required. Table 24-7 shows examples of general values of WCLK and <CNT>.

Table 24-7 General erase time for WCLK and <CNT>

	WCLK	8.33MHz	10MHz	12MHz
<CNT>	Number of counts	Programming time (μs)		
00	250	30.0	25.0	20.8
01	300	36.0	30.0	25.0
Other than the above	350	-	35.0	29.2

-: Setting is prohibited.

24.1.7.11 FCERASECR (Count Setting Register for Erase)

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

Bit	Bit Symbol	Type	Function
31-4	−	R	Read as "0".
3-0	CNT	R/W	Number of counts until erase time (CNT/WCLK) will be 100 to 130ms using each auto erase command 0000: Number of counts 850000 0001: Number of counts 900000 0010: Number of counts 950000 0011: Number of counts 1000000 0100: Number of counts 1050000 0101: Number of counts 1100000 0110: Number of counts 1150000 0111: Number of counts 1200000 1000: Number of counts 1250000 1001: Number of counts 1300000 1010: Number of counts 1350000 Other than the above: Number of counts 1400000

Note: If WCLKCR<DIV> is re-set, set a erase time within 100 to130 ms using auto erase command if required. Table 24-8 shows an example of general value of WCLK and <CNT>.

Table 24-8 General erase time for WCLK and <CNT>

	WCLK	8.33MHz	10MHz	12MHz
<CNT>	Number of counts	Erase time (ms)		
0000	850000	102.0	-	-
0001	900000	108.0	-	-
0010	950000	114.0	-	-
0011	1000000	120.0	-	-
0100	1050000	126.1	105.0	-
0101	1100000	-	110.0	-
0110	1150000	-	115.0	-
0111	1200000	-	120.0	-
1000	1250000	-	125.0	104.2
1001	1300000	-	-	108.3
1010	1350000	-	-	112.5
Other than the above	1400000	-	-	116.7

-: Setting is prohibited.

24.2 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.

Furthermore, for example while a program is executing on area 0, the other area (such as area 1) of Flash memory, on which instructions are not executed, can be written/erased. (The opposite case is possible.)

24.2.1 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 24-9 Flash memory function

Main function	Description
Auto program	Writes data in units of 4 word (16 bytes) automatically.
Auto chip erase	Erases the entire area of Flash memory automatically.
Auto area erase	Erases a selected area automatically.
Auto block erase	Erases a selected block automatically.
Auto page erase	Erases a selected page automatically.
Write/erase protect	The write or erase operation can be prohibited.
Auto memory swap	Automatically performs swap/swap release/swap size designation.

24.2.2 Operation Mode of Flash Memory

Flash memory provides main 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, 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 finished, the operation mode returns to the read mode except the ID-Read command. For details of command execution, refer to "24.2.3 How to Execute Command".During the automatic operation, data read and instruction execution stored in Flash memory cannot be performed.

24.2.3 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 "24.2.6 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 FCPSR0<RDY_BSY> = "0" is set.When the automatic operation ends, FCPSR0<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, refer to "24.2.4 Abortion of Automatic Operation".

Notes on the command execution

1. The following operation are prohibited while auto operation.
   - All interrupt request
2. To recognize command, command sequencer need to be in the read mode before command starting. Confirm if FCPSR0<RDY_BSY> ="1" is set prior to the first bus write cycle of each command. It is recommended that the read command is consecutively executed.
3. Command sequences must be executed on the on-chip RAM.
4. Set an area selection bit (write "111" (0x7) to <AREAn>) in FCAREASEL register before each command is executed.
5. Set each bus write cycle using consecutive 1-word (32-bit) data transfer instruction.
6. If an access is performed to the target Flash memory in each command sequence, an bus fault occurs.
7. When issuing commands, wrong addresses or data is written, make sure to issue software reset then return to the read mode.
8. Confirmation procedure after each command completion is as follow.
   1) Execute the final bus write cycle.
   2) Wait for FCPSR0<RDY_BSY> to become read "0" (Busy) by polling.
   3) Wait for FCPSR0<RDY_BSY> to become read "1" (Ready) by polling.
9. When a data is read from Flash memory, clear the area selection bit in FCAREASEL register. (Write "000" (0x0) to <AREAn>)

24.2.4 Abortion of Automatic Operation

The following procedure shows how to abort the automatic program/erase:

1. Read FCPSR0<RDY BSY>.
2. If the result of Procedure 1 is "1"(Ready), end the whole procedure at Procedure 9. If the result is "0", perform the following procedure at Procedure 3.
3. Write "0x7" to FCCR<WEABORT>.
4. Write "0x0" to FCCR<WEABORT>.
5. Poll until FCPSR0<RDY_BSY> is set to "1" (Ready).
6. Read FCSR<WEABORT>.

7. Issue a Read/reset command.

8. If the result of Procedure 6 is "0" (Note), end the whole procedure at Procedure 9. If the result is "1", perform the following procedure to clear FCPSR0<RDY_BSY>.
   (1) Write "0x7" to FCSTSCLR<WEABORT>.
   (2) Write "0x0" to FCSTSCLR<WEABORT>.
   (3) Poll until FCSR<WEABORT>="0".

9. End

Note: It is assumed that FCPSR0<RDY_BSY> = "1" is prior to Procedure 3.

24.2.5 Completion Notice of Automatic Operation

TMPM475FDFG/FZFG/FYFG provides the interrupt function that detects a completion notice of Flash programming/erase operation.

24.2.5.1 Procedure

The following is a procedure of how to set a completion notice of automatic operation.

For details of interrupt service routines, refer to "Interrupts" in Chapter "Exception".

1. After a programming/erase command is issued to Flash, check the automatic operation state (BUSY state) by FCPSR0<RDY_BSY>. After the confirmation of automatic operation, enable INTFLRDY interrupt.

2. After Flash automatic operation has been complete, INTFLRDY interrupt occurs.

3. In the INTFLRDY interrupt service routine, disable the event of INTFLRDY interrupt.

24.2.6 Command Description

This section explains each command content. For details of specific command sequences, refer to "24.2.7 Command Sequence".

24.2.6.1 Automatic Page Program

(1) Operation Description

The automatic page program writes data in the units of 4-word (16 bytes). 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 those 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.

Another command sequence is not accepted during automatic program. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

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 the units of one-word (32-bit).

If a part of 16 bytes needs to be written, write "0xFFFFFFFF" of 16 bytes as data which is not required to write.

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.

24.2.6.2 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 and return to the read mode after a command sequence is input.

Another command sequence is not accepted during automatic program. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

(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.

24.2.6.3 Automatic Area Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified area. If the specified area contains protected blocks, erase operation is not executed and returns to the read mode after a command sequence is input.

Another command sequence is not accepted during automatic erase operation. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

(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.

24.2.6.4 Automatic Block Erase

(1) Operation Description

The automatic erase command performs erase operation to the specified block. If the specified block contains protected pages or blocks, erase operation is not executed and returns to the read mode after a command sequence is input.

Another command sequence is not accepted during automatic erase operation. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

(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.

24.2.6.5 Automatic Page Erase

(1) Operation Description

The automatic page command performs erase operation to the specified page. If the specified page is protected, erase operation is not executed and returns to the read mode after a command sequence is input.

Another command sequence is not accepted during automatic erase operation. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

(2) How to Set

The 1st to 5th bus write cycles indicate the automatic page erase command sequence. In the 6th bus write cycle, the page to be erased in the bus write cycle is specified. After the command sequence is input, the automatic page 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.

24.2.6.6 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 "24.1.5 Protect/Security Function".

Another command sequence is not accepted during automatic erase operation. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

(2) How to Set

The 1st to 3th bus write cycles indicate the automatic protect bit program command sequence. In the 4th 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 protect bits in each FCPSR register, are written properly.

24.2.6.7 Automatic Protect Bit Erase

(1) Operation Description

The automatic protect bit erase command operation depends on the security status on the executions.

- Non-security status

  Clear all specified protect bits to "0".
- Security status

  Erase all protect bits after all addresses of Flash memory are erased.

For detail of security status, refer to "24.1.5 Protect/Security Function".

Another command sequence is not accepted during automatic erase operation. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, data writing may not be executed normally. Thus, automatic program must be executed again after erase operation.

(2) How to Set

The 1st to 3th bus write cycles indicate the automatic protect bit erase sequence. In the 4th bus write cycle, specify 0x5E000000. After the command sequence is input, the automatic protect bit erase operation starts.

In the non-security status, all protect bits are erased. Check whether the protect bits in each FCPSR register are erased properly.

In the security status, all addresses and all protect of Flash memory bits are erased. Confirm if data and protect bits are erased normally, and then execute the automatic protect bit erase, automatic chip erase or automatic block erase again if necessary.

24.2.6.8 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 sequence. In the 4th bus write cycle, the code to be read is specified. After the 4th bus write cycle, a code can be acquired by read operation in the arbitrary Flash area.

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/reset command.

24.2.6.9 Read/Reset Command (Software Reset)

(1) Operation Description

The read/reset command is a command to return Flash memory to the read mode.

When the ID-Read command is executed, Flash memory stops at the current status without automatically returning to the read mode. To return to the read mode from this situation, use the 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 sequence. After either command sequence is executed, Flash memory returns to the read mode.

24.2.6.10 Automatic Memory Swap

(1) Operation Description

Automatic memory swap is a command to write "1" to each bit of FCSWPSR[10:0] in the units of 1-bit. A bit cannot be set to "0" independently. All bits should be cleared to "0" using automatic protect bit erase command.

Another command sequence is not accepted during automatic memory swap operation. Refer to "24.2.4 Abortion of Automatic Operation" to stop this operation. At this time, this operation may not be executed normally. Thus, automatic memory swap must be executed again.

(2) How to Set

The 1st to 4th bus write cycles indicate the automatic memory swap command sequence. After the command sequence is input, "1" is written to the specified bit of FCSWPSR register.

24.2.7 Command Sequence

24.2.7.1 Command Sequence List

Table 24-10 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 24-11. Use below values to "command" described in a column of Addr[15:9] in the Table 24-11.

Note1) Use the following values for address bit [20:19] depending on the intended area.
Area0 : "00"

Note2) When the MCU moved from the STOP2 mode to the Normal mode, checking the CGRSTFLG<OSFLF> which is "1" are required for the Flash programming.

Table 24-10 Flash memory access by internal CPU

Command sequence	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/reset	0xXXXXXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
ID-Read	0xXX54XX	0xXXAAXX	0xXX54XX	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic program	0xXX54XX	0xXXAAXX	0xXX54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0xXX54XX	0xXXAAXX	0xXX54XX	0xXX54XX	0xXXAAXX	0xXX54XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Automatic area erase	0xXX54XX	0xXXAAXX	0xXX54XX	0xXX54XX	0xXXAAXX	AA	–
	0xAA	0x55	0x80	0xAA	0x55	0x20	–
Automatic block erase	0xXX54XX	0xXXAAXX	0xXX54XX	0xXX54XX	0xXXAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Automatic page erase	0xXX54XX	0xXXAAXX	0xXX54XX	0xXX54XX	0xXXAAXX	PGA	–
	0xAA	0x55	0x80	0xAA	0x55	0x40	–
Automatic protect bit program	0xXX54XX	0xXXAAXX	0xXX54XX	PBA	–	–	–
	0xAA	0x55	0x9A	0x9A	–	–	–
Automatic protect bit erase	0xXX54XX	0xXXAAXX	0xXX54XX	0x0000XX	–	–	–
	0xAA	0x55	0x6A	0x6A	–	–	–
Automatic memory swap	0xXX54XX	0xXXAAXX	0xXX54XX	MSA	–	–	–
	0xAA	0x55	0x9A	0x9A	–	–	–

Supplementary explanation

- IA: ID Address
- ID: ID Data
- PA: Program Address
- PD: Program Data (32-bit data)

After the 4th bus cycle, input 16 bits data in the order of the addresses

- AA: Area Address (see Table 24-2)
- BA: Block Address (see Table 24-2)
- PGA: Page Address
- PBA: Protect Bit Address (see Table 24-12)
- MSA: Memory Swap Address (see Table 24-14)

24.2.7.2 Address Bit Configuration in the Bus Cycle

Use Table 24-11 in conjunction with "Table 24-10 Flash memory access by internal CPU".

Set an address according to the normal bus write cycle address configuration in the first bus cycle and later.

Table 24-11 Address bit configuration in the bus write cycle

[Normal command]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:16]	Addr [15:9]	Addr [8:0]
Normal command	Setting of the bus write cycle address for normal command					
	0x5E	"000"	Note	"0" is recommended.	Command	"0" is recommended.

[Read/reset and ID-READ]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:16]	Addr [15:14]	Addr [13:0]
Read /reset	Setting of the 1st bus write cycle address for READ/Reset					
	0x5E	"000"	Fixed to "00".	"0" is recommended.		
ID-READ	IA: ID address (Setting of the 4th bus write cycle address for ID-READ)					
	0x5E	"000"	Fixed to "00".	"0" is recom-mended.	ID Address (Table 24-13)	"0" is recommended.

[Automatic area erase]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:0]
Area erase	AA: Area Address (Setting of the 6th bus write cycle address for area erase)			
	0x5E	"000"	Note	"0" is recommended.

[Automatic block erase]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:13]	Addr [12:0]
Block erase	BA: Block Address (Setting of the 6th bus write cycle address for block erase)				
	0x5E	"000"	Note	Block address (Table 24-2)	"0" is recommended.

[Automatic page erase]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:12]	Addr [11:0]
Page erase	PGA: Page Address (Setting of the 6th bus write cycle address for page erase)				
	0x5E	"000"	Note	Page address	"0" is recommended.

[Automatic program]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:3]	Addr [2:0]
Program	PA: Program page Address (Setting of the 4th bus write cycle address for page program)				
	0x5E	"000"	Note	Page Address	"0" is recom-mended.

[Protect bit program and memory swap]

Address	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:12]	Addr [11:8]	Addr [7:4]	Addr [3:0]
Protect bit erase	Setting of the 4th bus write cycle address for protect bit erase						
	0x5E	"000"	Fixed to "0".			"0" is recommended.	
Protect bit program	PBA: Protect bit address (Setting of the 4th bus write cycle address for protect bit program)						
	0x5E	"000"	Fixed to "00".	"0" is recommended.	Protect bit selection (Table 24-12)		"0" is recom-mended.
Memory swap	MSA: Memory Swap Address (Setting of the 4th bus write cycle address for memory swap)						
	0x5E	"000"	Fixed to "00".	"0" is recommended.	Memory swap bit selection (Table 24-14)		"0" is recom-mended.

Note) Use the following values for address bit [20:19] depending on the intended area.
Area0 : "00"

24.2.7.3 Area Address (AA) and Block Address (BA)

Table 24-2 shows area and block addresses. Specify any address included in the area/block to be erased in the 6th bus write cycle of the automatic block erase command. In the single chip mode, specify using the address of mirror area.

24.2.7.4 How to Specify Protect Bit (PBA)

A protect bit is specified in the units of 1-bit in operation.

Table 24-12 shows a protect bit selection table of the automatic protect bit program.

Table 24-12 Protect bit program address

Block	Page	Register	Protect bit	PBA[11:4]							Example of address [31:0]
				Address [11:10]	Address [9]	Address [8]	Address [7]	Address [6]	Address [5]	Address [4]	
0	0	FCPSR0	<PG0>	0	0	0	0	0	0	0	0x5E00_0000
	1		<PG1>	0	0	0	0	0	0	1	0x5E00_0010
	2		<PG2>	0	0	0	0	0	1	0	0x5E00_0020
	3		<PG3>	0	0	0	0	0	1	1	0x5E00_0030
	4		<PG4>	0	0	0	0	1	0	0	0x5E00_0040
	5		<PG5>	0	0	0	0	1	0	1	0x5E00_0050
	6		<PG6>	0	0	0	0	1	1	0	0x5E00_0060
	7		<PG7>	0	0	0	0	1	1	1	0x5E00_0070
1	8 to 15		<BLK1>	0	0	0	1	0	0	0	0x5E00_0080
2	16 to 23		<BLK2>	0	0	0	1	0	0	1	0x5E00_0090
3	24 to 31		<BLK3>	0	0	0	1	0	1	0	0x5E00_00A0
4	32 to 39		<BLK4>	0	0	0	1	0	1	1	0x5E00_00B0
5	40 to 47		<BLK5>	0	0	0	1	1	0	0	0x5E00_00C0
6	48 to 55		<BLK6>	0	0	0	1	1	0	1	0x5E00_00D0
7	56 to 63		<BLK7>	0	0	0	1	1	1	0	0x5E00_00E0
8	64 to 71		<BLK8>	0	0	0	1	1	1	1	0x5E00_00F0
9	72 to 79		<BLK9>	0	0	1	0	0	0	0	0x5E00_0100
10	80 to 87		<BLK10>	0	0	1	0	0	0	1	0x5E00_0110
11	88 to 95		<BLK11>	0	0	1	0	0	1	0	0x5E00_0120
12	96 to 103		<BLK12>	0	0	1	0	0	1	1	0x5E00_0130
13	104 to 111		<BLK13>	0	0	1	0	1	0	0	0x5E00_0140
14	112 to 119		<BLK14>	0	0	1	0	1	0	1	0x5E00_0150
15	120 to 127		<BLK15>	0	0	1	0	1	1	0	0x5E00_0160

24.2.7.5 ID-Read Command (IA, ID)

Table 24-13 shows how to specify a code and the content using ID-Read command.

Table 24-13 ID-Read Command codes and contents

Code	ID[7:0]	IA[15:14]	Example of address [31:0]
Manufacture code	0x0098	00	0x5E00_0000
Device code	0x005A	01	0x5E00_4000
-	Reserved	10	-
Macro code	0x012F	11	0x5E00_C000

24.2.7.6 Memory Swap Bit Assignment (MSA)

Table 24-14 shows setting values of FCSWPSR[10:0] assigned in the 4th bus write cycle of auto memory swap command.

Table 24-14 Setting values of FCSWPSR[10:0] by memory swap command and example of address

FCSWPSR[10:0]	MSA[11:4]							Example of address [31:0]
	Address [11]	Address [10:9]	Address [8]	Address [7]	Address [6]	Address [5]	Address [4]	
FCSWPSR[0]	1	Fixed to "0".	0	1	0	0	0	0x5E00_0880
FCSWPSR[1]	1	Fixed to "0".	0	1	0	0	1	0x5E00_0890
FCSWPSR[2]	1	Fixed to "0".	0	1	0	1	0	0x5E00_08A0
FCSWPSR[3]	1	Fixed to "0".	0	1	0	1	1	0x5E00_08B0
FCSWPSR[4]	1	Fixed to "0".	0	1	1	0	0	0x5E00_08C0
FCSWPSR[5]	1	Fixed to "0".	0	1	1	0	1	0x5E00_08D0
FCSWPSR[6]	1	Fixed to "0".	0	1	1	1	0	0x5E00_08E0
FCSWPSR[7]	1	Fixed to "0".	0	1	1	1	1	0x5E00_08F0
FCSWPSR[8]	1	Fixed to "0".	1	0	0	0	0	0x5E00_0900
FCSWPSR[9]	1	Fixed to "0".	1	0	0	0	1	0x5E00_0910
FCSWPSR[10]	1	Fixed to "0".	1	0	0	1	0	0x5E00_0920

24.2.7.7 Example of Command Sequence

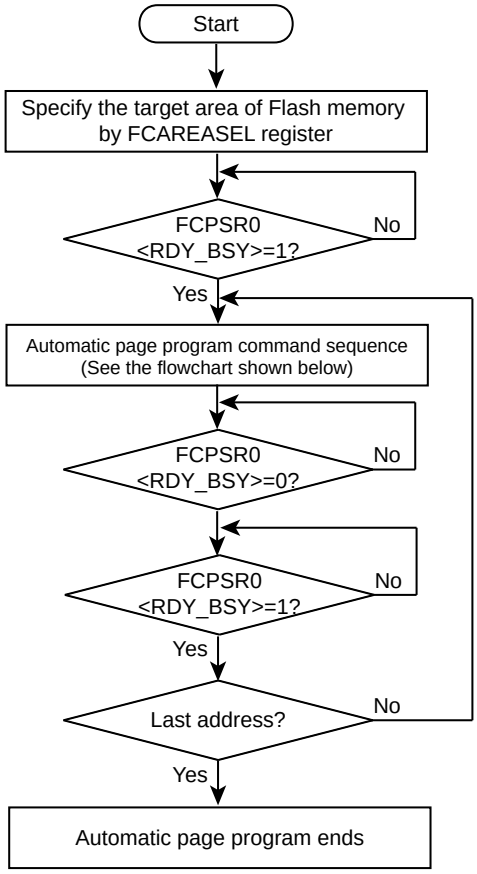
Table 24-15 Example of Command Sequence

Command		Bus cycle						
		1	2	3	4	5	6	7
Read/Reset	Address	0x5E00_0000	–	–	–	–	–	–
	Data	0x0000_00F0	–	–	–	–	–	–
ID-Read	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	IA	0x5E00_0000	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID	–	–
Auto chip erase	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	0x5E00_5400	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	–
Auto protect bit program	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	PBA	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A	–	–	–
Auto protect bit erase	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_0000	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A	–	–	–
Auto memory swap	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	MSA	–	–	–
	Data	0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_009A	–	–	–

Command		Bus write						
		1	2	3	4	5	6	7
Automatic program	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	PA	Hereafter,16 bits data are consecutively written.		
	Data	0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
Automatic area erase	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	0x5E00_0000	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0020	–
Automatic block erase	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	BA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	–
Automatic page erase	Address	0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	PGA	–
	Data	0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0040	–

24.2.8 Flowchart

24.2.8.1 Automatic Program



Automatic page program command sequence (Address/command)

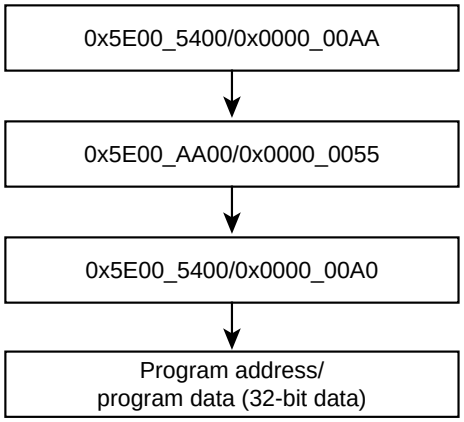


Figure 24-5 Flowchart of Automatic Program

24.2.8.2 Automatic Erase

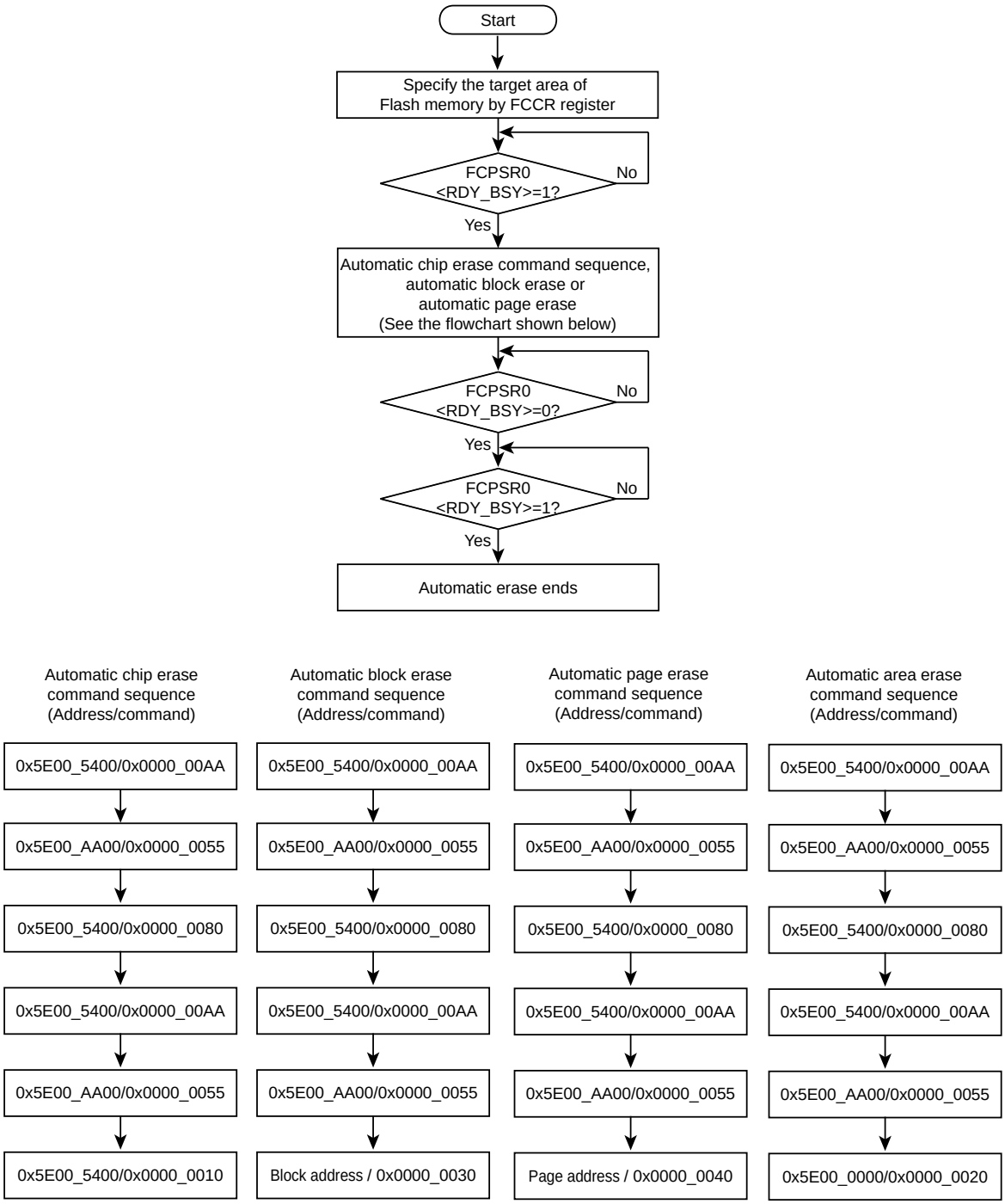


Figure 24-6 Flowchart of automatic erase

24.3 How to Reprogram Flash in 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 single boot mode, Flash memory is reprogrammed using serial command/data transfer. With connecting serial channel (SIO/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 exception to avoid abnormal program termination.

To secure the contents of Flash memory in the single chip mode (normal operation mode), it is recommended to protect relevant Flash blocks against accidental erasure during subsequent single chip operations after re-programming is complete.

24.3.1 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.

$\overline{BOOT}$ = 0
$\overline{RESET}$ = 0 → 1

While $\overline{BOOT}$ pin is set to the above conditions in advance, set $\overline{RESET}$ pin to "0". Then release $\overline{RESET}$ pin to boot-up in the single boot mode.

24.3.2 Interface Specification

This section describes the serial communication format in the single boot mode. The serial operation supports UART (asynchronous communication) mode. In order to execute the on-board programming, set the communication format of the programming controller as well.

Communication channel:	channel 0
Serial transfer mode:	UART (asynchronous), half-duplex, LSB first
Data length:	8-bit
Parity bit:	Note
STOP bit:	1-bit
Baud rate:	Arbitrary baud rate

The internal boot program operates on 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 "24.3.5.1 Serial Operation Mode Determination", a baud rate is determined by the 16-bit timer (TMRB). Since communications are executed by 1/16 of a desired baud rate when determining the baud rate; therefore the communication baud rate must be within the measurable range. The timer count clock operates at ΦT1(fc/2).

Table 24-16 shows the pins used in the boot program. Other than these pins are not used by the boot program.

Table 24-16 Used Pin

Mode setting pin	$\overline{\text{BOOT}}$(PF0)
Reset pin	$\overline{\text{RESET}}$
Communication pin	SC0TXD (PE0)
	SC0RXD (PE1)

24.3.3 Restrictions on Built-in Memories

Note that the single boot mode places restrictions on the built-in RAM and built-in flash memory as shown in Table 24-17.

Table 24-17 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.
Internal Flash memory	The following addresses are assigned for storing software ID information and passwords. Should not use the following address for program storage. 0x5E07_FFF0 to 0x5E07_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

24.3.4 Operation Command

The boot program provides the following operation commands

Table 24-18 Operation command data

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory chip erase and protect bit erase

24.3.4.1 RAM Transfer

The RAM transfer is to store data from the controller to the internal 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 at RAM store start 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 24.2.7.

24.3.4.2 Flash Memory Chip Erase and Protect Bit Erase

"Flash memory chip erase and protect bit erase command" erases the entire blocks of Flash memory. This command erases all blocks regardless of write/erase protect condition or security status.

24.3.5 Common Operation regardless of Command

This section describes common operation under the boot program execution.

24.3.5.1 Serial Operation Mode Determination

The controller sends 0x86 of the 1st byte at the desired baud rate. Figure 32-6 shows waveforms in each case.

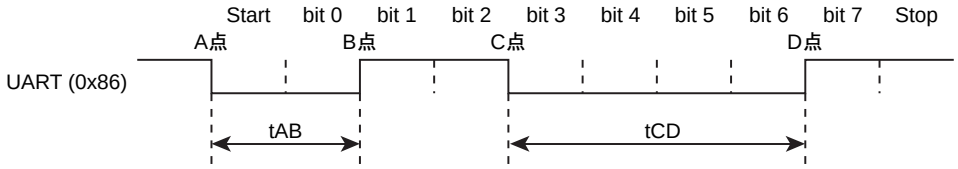


Figure 24-7 Serial operation mode determination data

Figure 24-8 shows a flow chart of internal 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 24-8, the CPU monitors the level of the receive pin, and obtains a timer value at the moment when the receive pin's level is changed. Therefore, the timer values of tAB, tAC and tAD have a margin of error. In addition, note that if a transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin.

The flow chart in Figure 24-9 shows the serial operation mode is determined by whether 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 for whether the automatic baud rate setting is enable or not. If the length is tAB > tCD, the serial operation mode is not determined as UART Interface mode. Note that timer values of tAB, tAC and tAD have a margin of error. If the baud rate is high but operation frequency is low, each timer value becomes small. This may generate unexpected determination. (To prevent this problem, re-set UART within the programming routine.)

For example, the serial operation mode may not be determined to be UART Interface mode when the controller attempts to use UART mode. To avoid such situation, 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. If it fails to obtain that echo-back within the allowed time, the controller is incapable of communications.

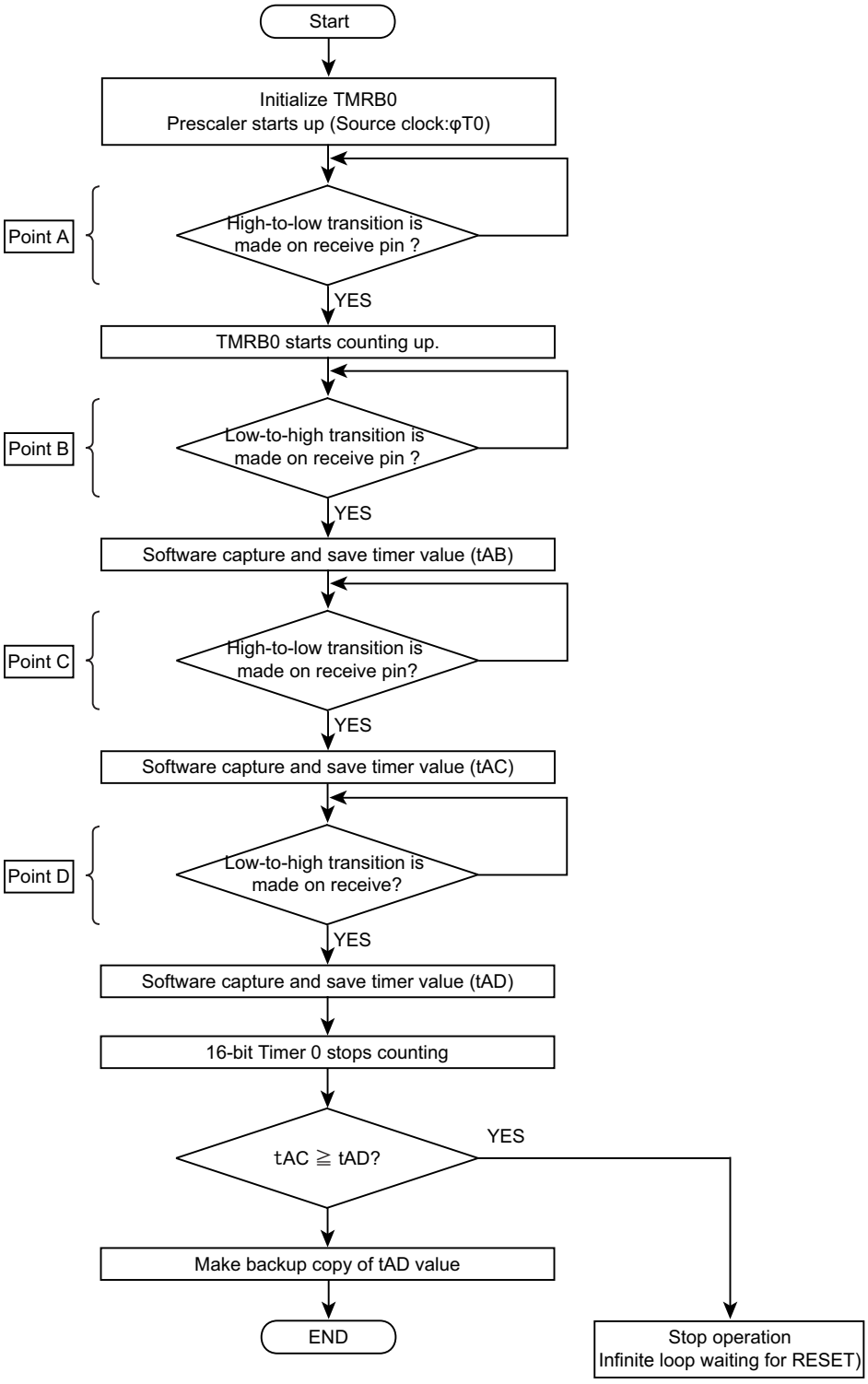


Figure 24-8 Serial operation mode receive flow chart

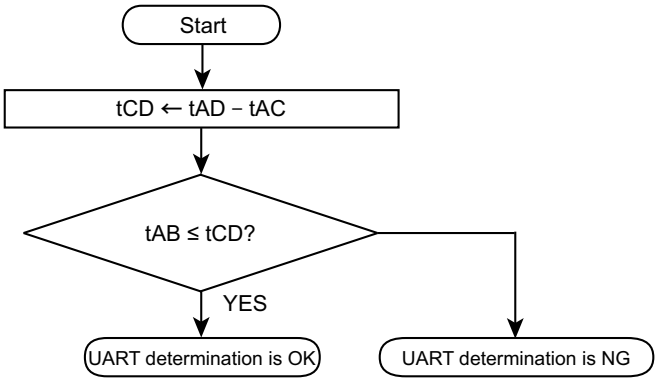


Figure 24-9 Serial operation mode determination flow chart

24.3.5.2 Acknowledge Response Data

The internal boot program represents processing states in specific codes and sends them to the controller. Table 24-19 to Table 24-22 show the values of acknowledge responses to each receive data.

In Table 24-20 to Table 24-22, the upper four bits of the acknowledge response are equal to those of upper 4 bits 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".

Table 24-19 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 24-20 ACK response to the operation command data

Transmit data	Description
0x?8 (Note)	A receive error occurs in the operation command data.
0x?1 (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 24-21 ACK response to the CHECK SUM data

Transmit data	Description
0xN8 (Note)	A receive error occurred.
0xN1 (Note)	A CHECK SUM or password error occurred.
0xN0 (Note)	The CHECK SUM value was determined as correct.

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 24-22 ACK response to Flash memory chip erase and protect bit erase operation

Transmit data	Description
0x54	Determined as erase enable command.
0x4F	Erase command ended
0x4C	Erase command ended illegally.
0x47	Flash operation command was aborted.

24.3.5.3 Password Determination

The boot program uses the below area as the examination of necessity of password and password data area.

Area	Address
Examination of necessity of password	0x5E17_FFF0 (1byte)
Password	0x5E17_FFF4 to 0x5E17_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

This item describes the password determination described in No.5 of "Table 24-24 Communication rules of RAM transfer".

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 24-10. In this case, the boot program returns an error acknowledge (0x11) in response to the checksum value regardless of the password verification.

The boot program verifies receive data (password data). A password error occurs if all 12 bytes do not match. If the password error is determined, an ACK response becomes a password error.

The password verification is performed even when the security function is enabled.

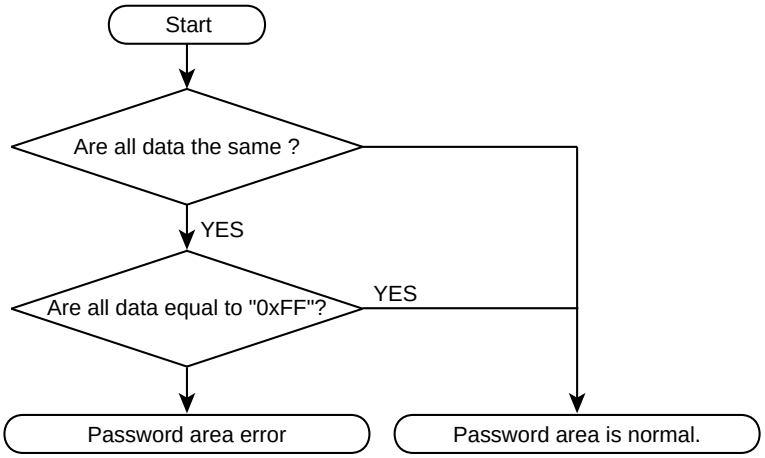


Figure 24-10 Password area check flow chart

(2) Password verification to Flash memory chip erase and protect bit erase command

This item describes the password determination described in No.5 of "Table 24-25 Communication Rules of Flash memory Chip Erase and Protect Bit Erase".

When a password is valid in the erase password necessity determination area as shown in Figure 24-11. If the passwords are identical data, a password area error occurs. If a password area error is determined, an ACK response to CHECK SUM value sends 0x41 regardless of the password verification.

The boot program verifies 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 CHECK SUM data is a password error. The password verification is performed even when the security function is enabled.

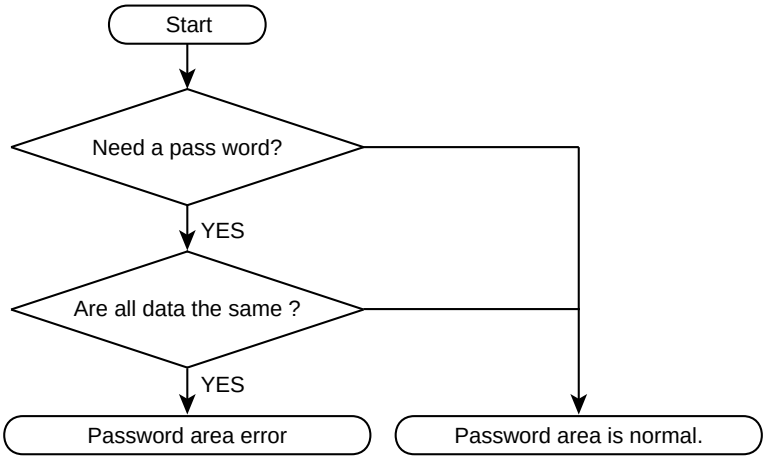


Figure 24-11 Password area check flow chart

24.3.5.4 CHECK SUM Calculation

The CHECK SUM is calculated by 8-bit addition (ignoring the overflow) to transmit data and taking the two's complement of the sum of lower 8 bits. Use this calculation when the controller transmits the CHECK SUM value.

Example calculation of CHECK SUM

To calculate the CHECK SUM for 2 bytes data (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

24.3.6 Communication Rules for Determination of Serial Operation Mode

This section describes the communication rule for determination of serial operation mode. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM475

Transfer direction (C←T): TMPM475 to Controller

Table 24-23 Communication rules for determination of serial operation mode

No	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Controller transmits data to determine the serial operation mode. For details of mode determination of the target, refer to "24.3.5.1 Serial Operation Mode Determination".
		0x86	Controller transmits 0x86. If the target determines UART mode is OK, the target successively determines whether baud-rate setting is possible or not. If not, the program stops and communication is shutdown.
2	C←T	ACK response to serial operation mode	Receive data from the controller is the ACK response data responding to 1st byte of serial operation mode setting data. If the target determines the setting is possible, the target sets UART. A receive enable timing is set before transmit buffer is written to the data.
		Normal state: 0x86	If the target determines the setting is possible, the target transmits 0x86. If the target determines the setting is not possible, the target does not transmit anything and stops the operation. Set a timeout time (5 sec.) after the controller finished transmitting 1st byte of data. If the controller does not receive data (0x86) properly within the timeout time, it should be determined as a communication failure. If data (0x86) is not normally received within a time-out time, communications are not possible.
3	−	−	Controller transmits operation command data. For details of transfer format of each operation command, refer to "24.3.7 Communication Rules at RAM Transfer" or "24.3.8 Communication Rules of Flash memory Chip Erase and Protect Bit Erase".

24.3.7 Communication Rules at RAM Transfer

This section shows a communication rules of RAM transfer. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TMPM475

Transfer direction (C←T): TMPM475 to Controller

Table 24-24 Communication rules of RAM transfer

No	Transfer direction	Transfer data	Description
1	C→T	Operation command data (0x10)	Controller transmits RAM transfer command data (0x10).
2	C←T	ACK response to the operation command Normal: 0x10 Abnormal: 0x11 Communication error: 0x18	Target checks received data and sends ACK response data. If a receive error exists, the target responses ACK response data 0x18 indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks the data against operation command data described in Table 24-18. If checking is failed, the target responses ACK response data 0x11 indicating abnormal state, and then returns to the initial stat waiting for operation command data. If checking is succeeded, the target responses ACK response data 0x10 indicating normal state, and then it waits for next data.
3	C→T	Password data (12 bytes)	The controller transmits data which is the same area as the password data of Flash memory. For details of password data area, refer to "24.3.5.3 Password Determination".
4	C→T	CHECK SUM value of transmit data (No.3)	The controller transmits a CHECKSUM value of transmit data (No.3). For details of CHECK SUM calculation, refer to "24.3.5.4 CHECK SUM Calculation".
5	C←T	ACK response to CHECK SUM value Normal: 0x10 Abnormal: 0x11 Communication error: 0x18	The target checks receive data and responses ACK response data. If a receive error exists, the target responses ACK response data 0x18 indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECK SUM value and password. For details of password determination, refer to "24.3.5.3 Password Determination". If password determination is failed, the target responses ACK response data 0x11 indicating abnormal state, and then returns to the initial state waiting for operation command data. If password determination is succeeded, the target responses ACK response data 0x10 indicating normal state, and then it waits for next transmit data.
6	C→T	RAM store start address 31 to 24	Transmit the RAM start address to be stored in RAM store data by dividing into 4 times as a next transmit data from the controller. Transmission order is as follows:1st byte corresponds to 31 bit to 24 bit and 4th byte corresponds to 7th bit to 0th bit of transfer address. These addresses should be placed in 0x2000_0400 through the last address of RAM address. The target checks receive data. If a receive error exists, the target responses ACK response data 0x18 indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target does not transmit anything, and it waits for next transmit data.
7	C→T	RAM store start address 23 to 16	
8	C→T	RAM store start address 15 to 8	
9	C→T	RAM store start address 7 to 0	
10	C→T	Number of RAM store bytes 15 to 8	Transmit the number of byte to be block-transferred from the controller. Transmission order is as follows: 1st byte corresponds to 15 bit to 8 bit and 2nd byte corresponds to 7th bit to 0th bit of transfer address. These addresses should be placed in 0x2000_0400 through the last address of RAM address. The target checks receive data. If a receive error exists, the target responses ACK response data 0x18 indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target does not transmit anything, and it waits for next transmit data.
11	C→T	Number of RAM store byte 7 to 0	
12	C→T	CHECK SUM value of transmit data (No.6 to 11)	Transmit a CHECK SUM value of transmit data (No.6 to 11) from the controller.

No	Transfer direction	Transfer data	Description
13	C←T	ACK response to CHECK SUM value Normal: 0x10 Abnormal: 0x11 Communication error: 0x18	The target checks receive data and responses ACK response data. If a receive error exists, the target responses ACK response data 0x18 indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECK SUM value. If checking is failed, the target responses ACK response data 0x11 indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target responses ACK response data 0x10 indicating normal state, and then it waits for next data.
14	C→T	RAM stored data	Transmit data to be stored in RAM from the controller. The target receives data to be stored in RAM.
15	C→T	CHECK SUM value of transmit data (No. 14)	Transmit a CHECK SUM value of transmit data (No.14) from the controller.
16	C←T	ACK response to CHECK SUM value Normal:0x10 Abnormal: 0x11 Communication error: 0x18	The target checks receive data and responses ACK response data. If a receive error exists, the target responses ACK response data 0x18 indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECK SUM value. If checking is failed, the target responses ACK response data 0x11 indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target responses ACK response data 0x10 indicating normal state and jumps to RAM store start address (No.6 to 9) as a branch address.

24.3.8 Communication Rules of Flash memory Chip Erase and Protect Bit Erase

This section shows a communication 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 TMPM475

Transfer direction "C←T": TMPM475 to Controller

Table 24-25 Communication Rules of Flash memory Chip Erase and Protect Bit Erase

No	Transfer direction	Transfer data	Description
1	C→T	Operation command data (0x40)	Sends Flash memory chip erase and protect bit erase command data (0x40).
2	C←T	ACK response to operation command Normal: 0x40 Abnormal: 0x41 Communication error: 0x48	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 communication sand 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 24-13, receive data is echoed back. If the data does not correspond to the command in Table 24-13, 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.)
3	C→T	Password data (12 bytes)	Transmit data which is the same as password data area of Flash memory from the controller. However; if password requirement of Flash memory is set to "no" (data: 0xFF), the target does not conduct password verification, so that dummy data can be used as a password. For details of password data area, refer to "24.3.5.3 Password Determination".
4	C→T	CHECK SUM value of transmit data (No.3)	Transmit a CHECK SUM value of transmit data (No.3) from the controller. For a method of CHECK SUM calculation, refer to "24.3.5.4 CHECK SUM Calculation".
5	C←T	ACK response to CHECK SUM value Normal: 0x40 Abnormal: 0x41 Communication error: 0x48	The target checks receive data and responses ACK response data. If receive error exists, the target responses ACK response data 0x48 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If receive error does not exist, the target checks a CHECK SUM value. If checking is failed, the target responses ACK response data 0x41 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target performs password checking. If password requirement is set to "no", the target transmits ACK response data 0x40 indicating normal. If password requirement is set to "need password", the target checks the password. If password checking is failed, the target responses ACK response data 0x41 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If password checking is succeeded, the target responses ACK response data 0x40 indicating normal and then it waits next data.
6	C→T	Erase enable command data (0x54)	Transmit erase enable command data (0x54) from the controller.
7	C←T	ACK response to erase enable command Normal: 0x54 Abnormal: 0x51 Communication error: 0x58	The target checks receive data and responses ACK response data. If receive error exists, the target responses ACK response data 0x58 indicating communication error, and then returns to the initial state waiting for operation command data. If receive error does not exist, the target checks erase enable command (0x54). If checking is failed, the target responses ACK response data 0x51 indicating abnormal communication, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target responses ACK response data 0x54 indicating normal, and then chip erase process is performed.
8	C←T	ACK response to erase command Normal: 0x4F Abnormal: 0x4C Abort chip erase command: 0x47	The target responses the result of chip erase process. If any problems occur, the target responses ACK response data (0x4F) indicating normal. If an blank check error occurs, the target responses ACK response data (0x4C) indicating abnormal. If chip erase command is aborted, the target responses ACK response data (0x47) indicating abort and then returns to the initial state waiting for operation command data.

24.3.9 Boot Program Whole Flowchart

This section shows a boot program whole flow chart.

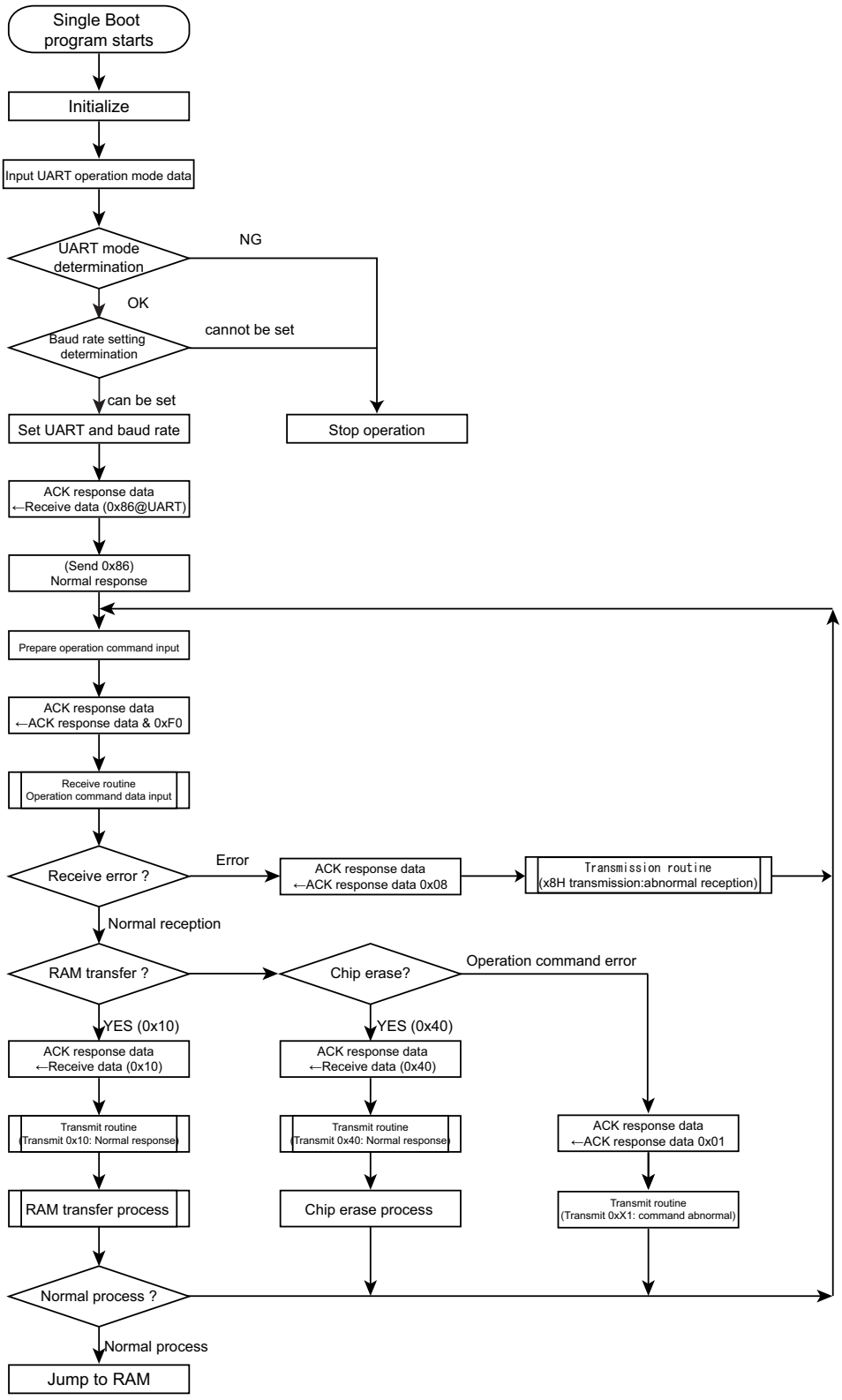


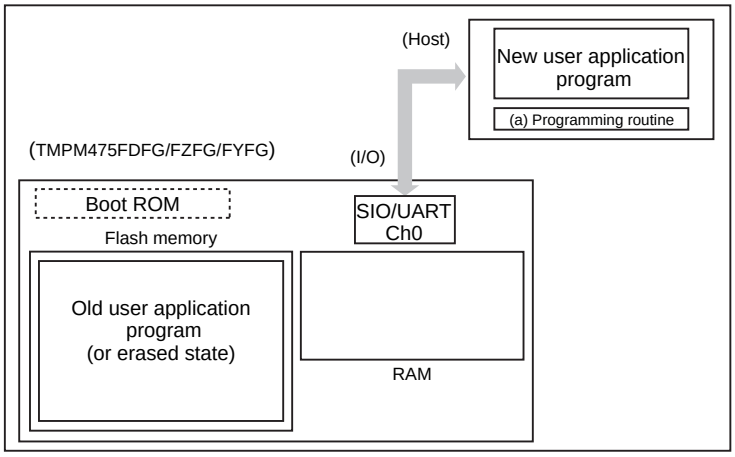
Figure 24-12 Boot Program Whole Flowchart

24.3.10 Reprogramming Procedure of Flash Using Reprogramming Algorithm in BOOT ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip boot ROM.(The Following example is using a ch0 of UART/SIO channel)

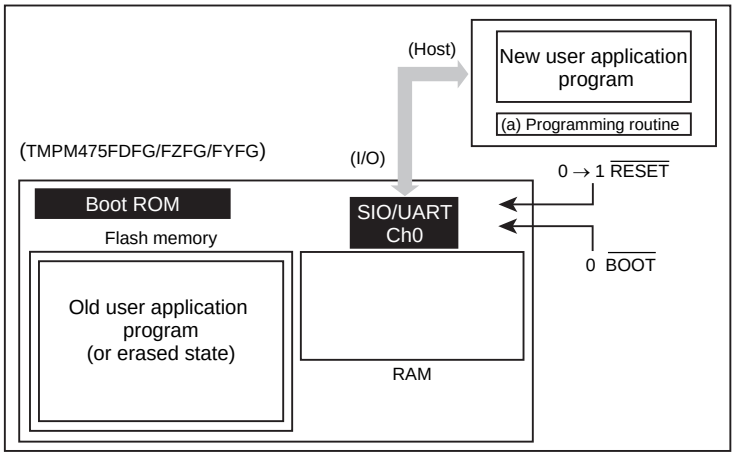
24.3.10.1 Step-1

The condition of Flash memory does not need to care whether a former user program 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.



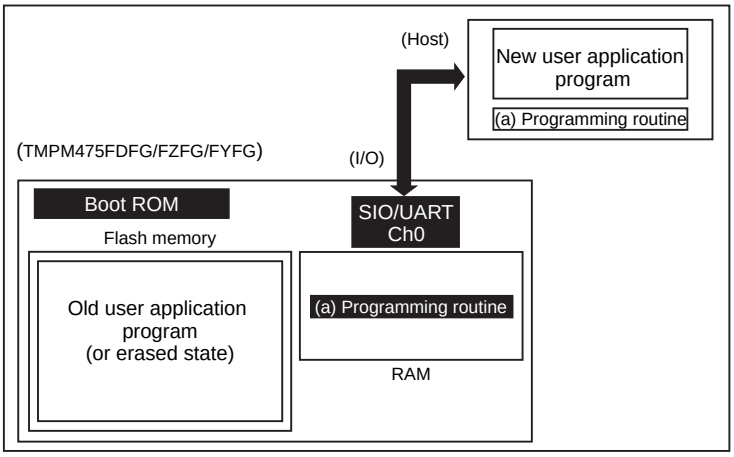
24.3.10.2 Step-2

Release the reset by the pin condition setting in the boot mode and boot-up on the BOOT ROM. According to the procedure of boot mode, transfer the programming routine (a) via SIO0 from the source (host). A password verification in the user application program is performed first. (If Flash memory is erased, erase data (0xFF) is dealt as a password.)



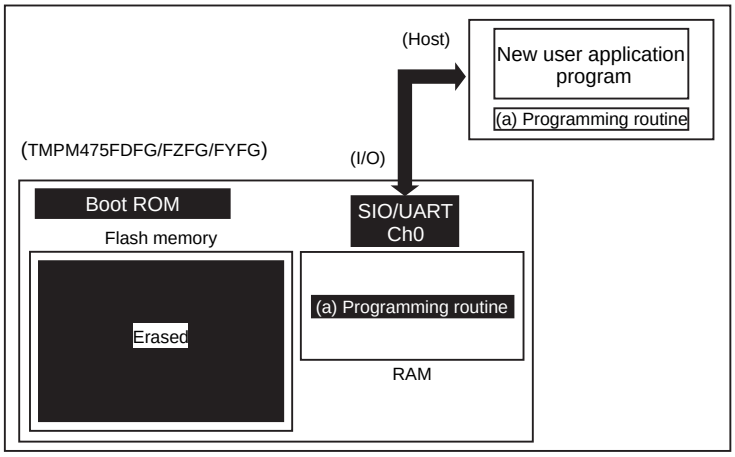
24.3.10.3 Step-3

If the password verification is complete, the boot program transfers 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.



24.3.10.4 Step-4

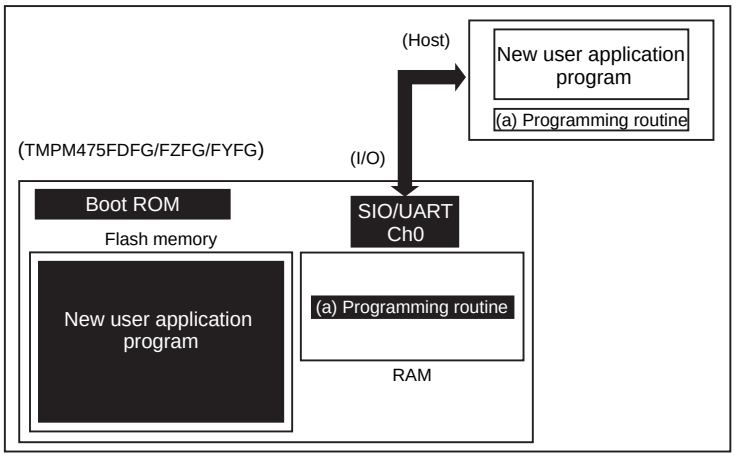
The boot program jumps to the programming routine (a) in the on-chip RAM to erase the Flash block containing old application program codes (The units of erase can be anything).



24.3.10.5 Step-5

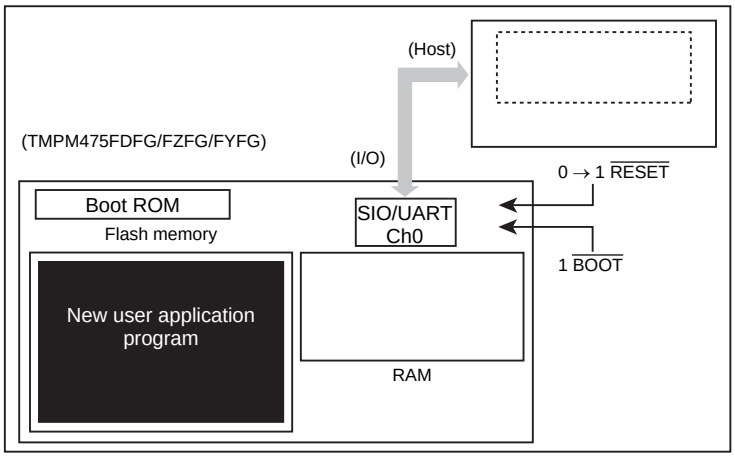
The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is complete, the writing or erase protection of that Flash area in the user's program must be ON.

In the example below, new program codes come 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.



24.3.10.6 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.



24.4 Reprogramming in the User Boot Mode

A user boot mode is to use Flash memory programming on the internal RAM of users' set. It is used when the data transfer bus for Flash memory program code on the user application is different from the serial I/O. It operates in the single chip mode; therefore, normal mode, in which user application is activated in the single chip mode, is required to switch to the user boot mode for programming Flash memory. Consequently, 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 conditions. 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. Once re-programming is complete, write/erase protection to the necessary block is recommended to avoid from accidental modification in the single chip mode (normal operation mode). Make sure not to generate exception to avoid abnormal termination even in the user boot mode.

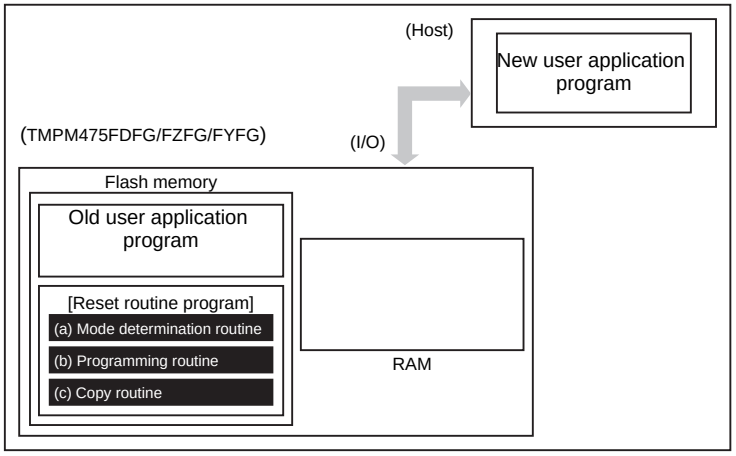
Taking examples from two cases such as the method that reprogramming routine stored in Flash memory (1-A) and transferred from the external device (1-B), the following section explains the procedure. For details of the program/erase Flash memory, refer to "24.2 Detail of Flash Memory".

24.4.1 (1-A) Procedure that a Programming Routine Stored in Flash memory

24.4.1.1 Step-1

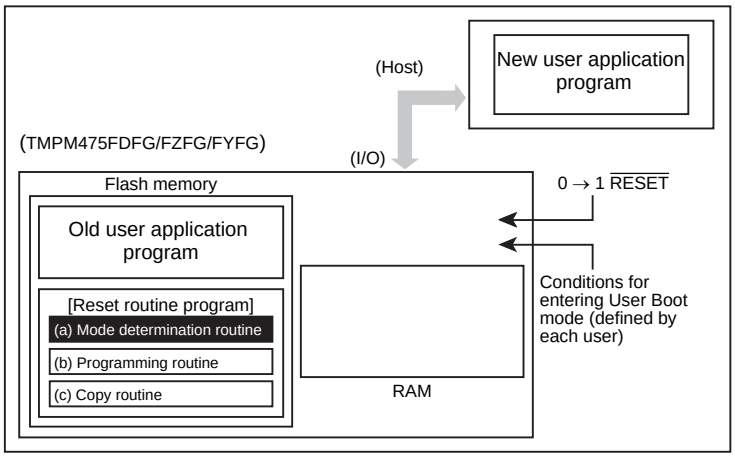
A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O bus to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, write the following three program routines into an arbitrary Flash 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 reprogramming routine:	A program to download new program from the external device and re-program Flash memory
(c) Copy routine:	A program to copy the data described in (a) to the built-in RAM.



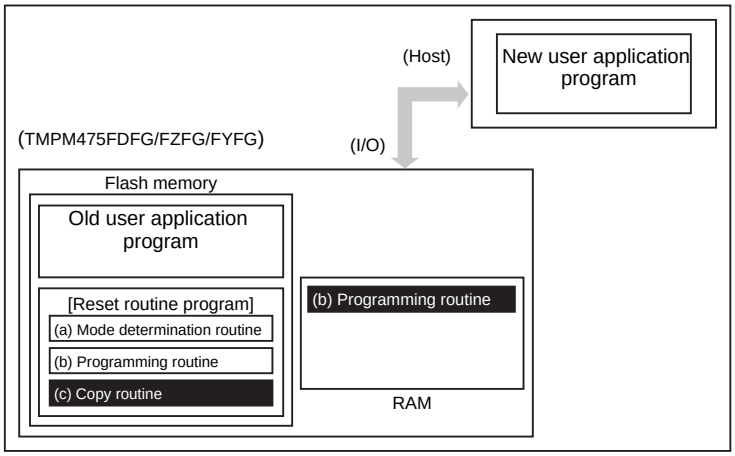
24.4.1.2 Step-2

This section explains the case that a programming routine stored 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. (Prohibit to generate all exception after enter in user boot mode)



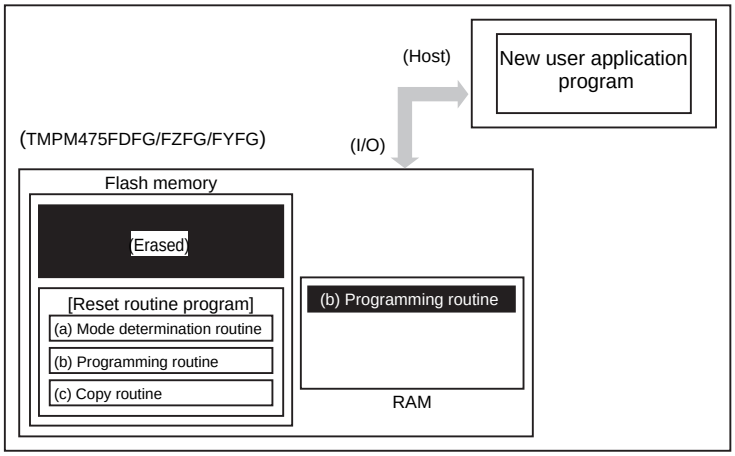
24.4.1.3 Step-3

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 internal RAM.



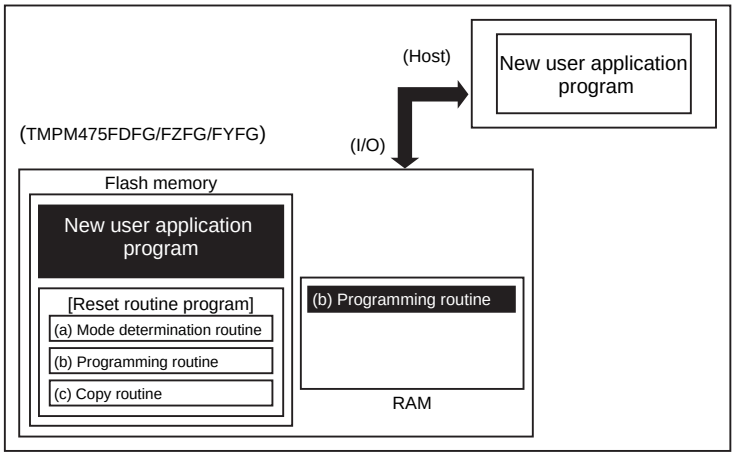
24.4.1.4 Step-4

Jump to the reprogramming routine on the RAM to release the write/erase protection for the old application program, and to erase Flash. (The units of erase can be anything)



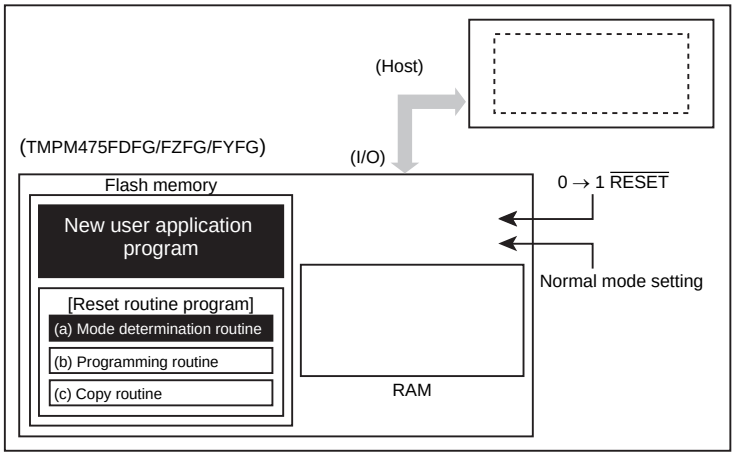
24.4.1.5 Step-5

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 ON.



24.4.1.6 Step-6

Do reset by setting "0" to $\overline{\text{RESET}}$. Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



24.4.2 (1-B) Procedure that a Programming Routine is transferred from External Host

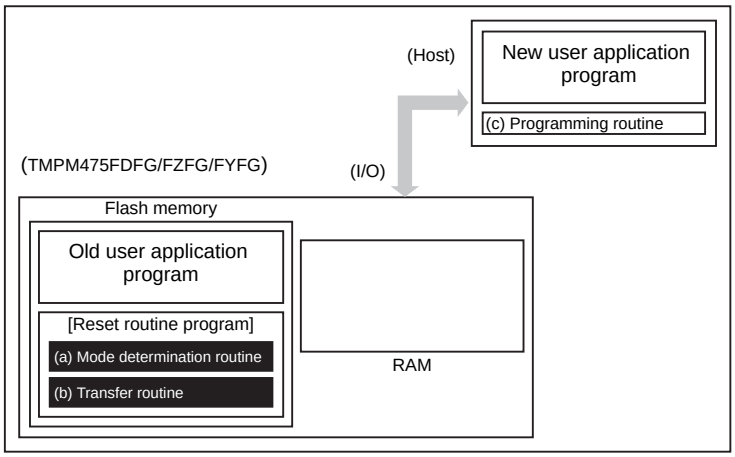
24.4.2.1 Step-1

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.

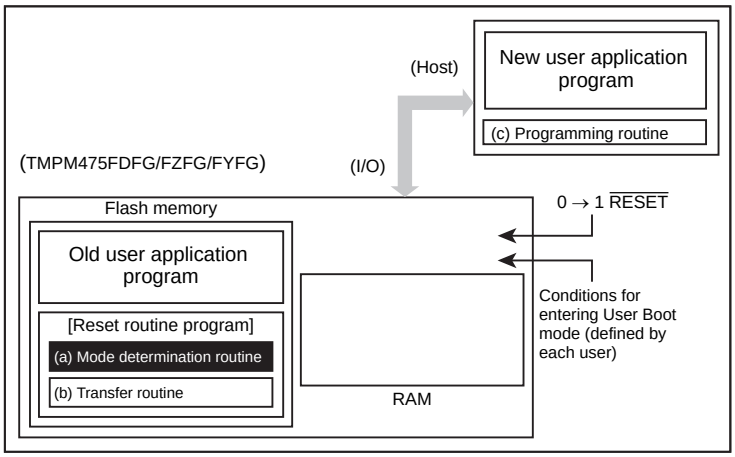
In addition, prepare a reprogramming routine shown below must be stored on the host controller.

(c) Reprogramming routine: A program to reprogram data



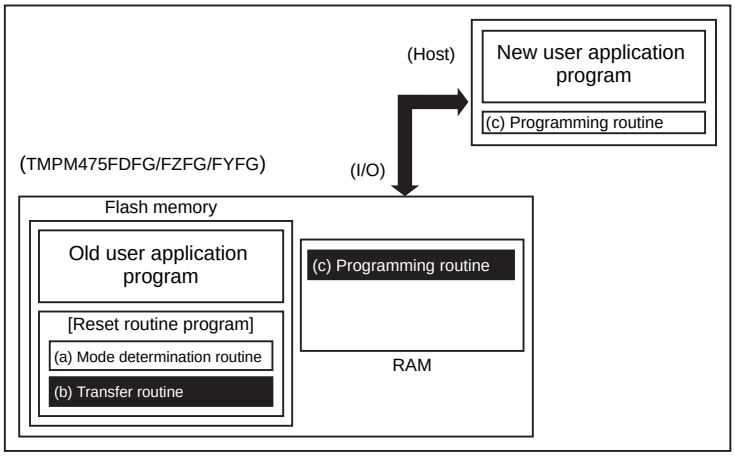
24.4.2.2 Step-2

This section explains the case that a programming routine stored 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. (Prohibit to generate all exception after enter in user boot mode)



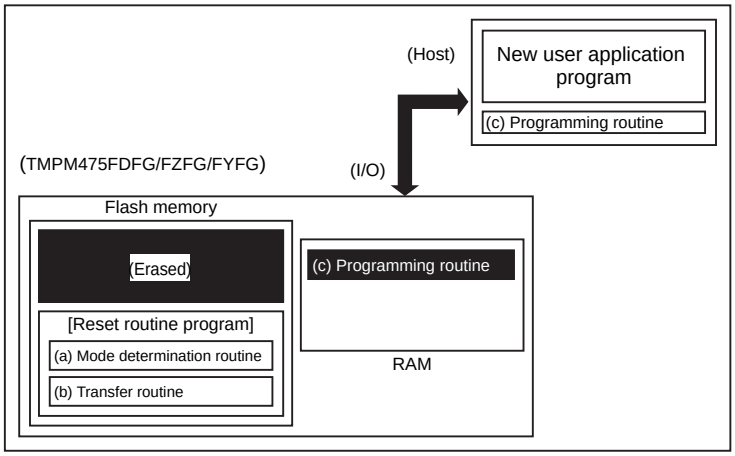
24.4.2.3 Step-3

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 internal RAM.



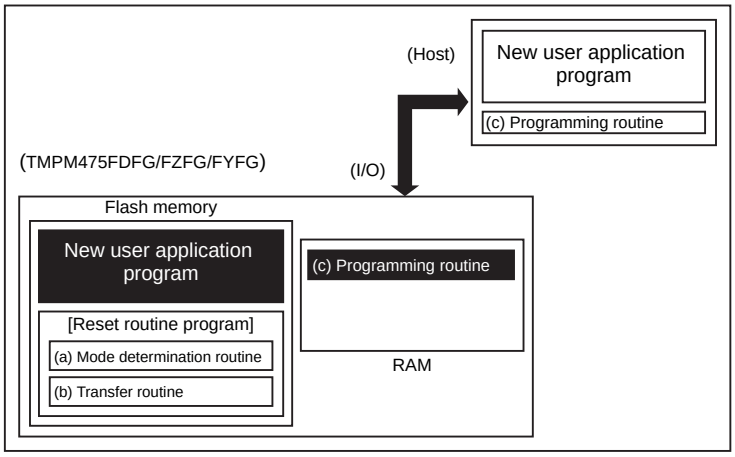
24.4.2.4 Step-4

Jump to the reprogramming routine in the internal RAM to release the write/erase protection for the old application program, and to erase Flash. (The units of erase can be anything)



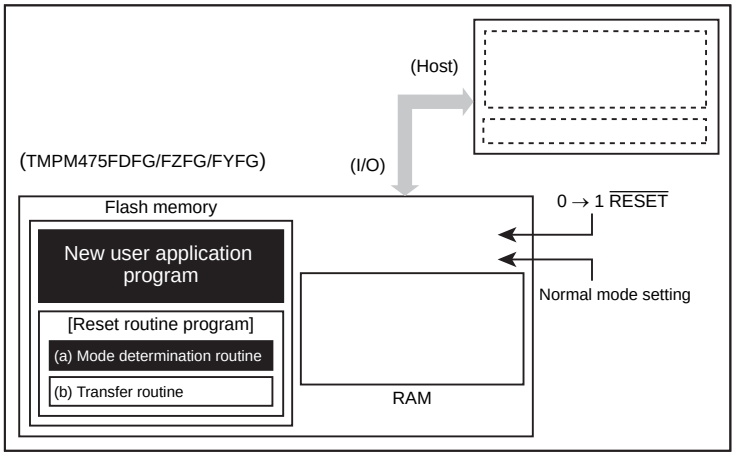
24.4.2.5 Step-5

Continue to execute the Flash programming routine (c) to download new program data from the host controller and program it into the erased Flash blocks. When the programming is complete, the write/erase protection of that Flash block in the user program area must be ON.



24.4.2.6 Step-6

Do reset by setting "0" to $\overline{\text{RESET}}$. Upon reset, Flash memory is set to the normal mode. After reset, the CPU will start along with the new application program.



24.5 How to Reprogram Flash using User Boot Mode

This method switches the Page 0 area to Page 1 area to leave a user boot program using swap function when Flash memory is reprogrammed.

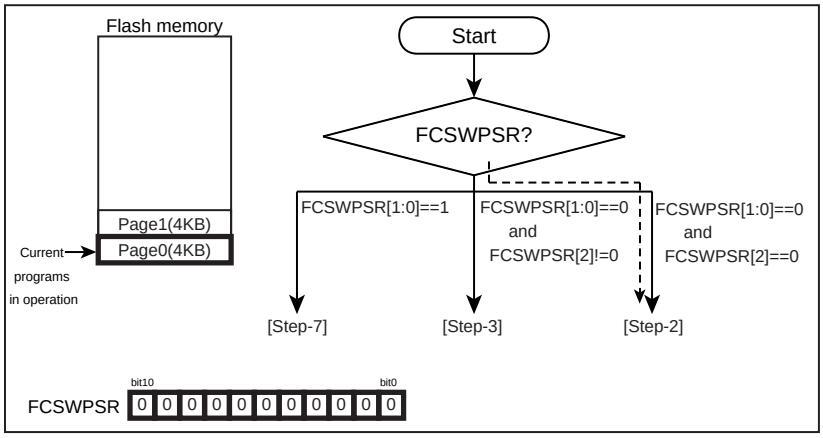
The following is an example of reprogramming procedure of user boot program.

(Assumed conditions in the following explanations: Swap size is 4K bytes. Page 1 program is copied from Page 0.)

24.5.1 Example of Flash Memory Reprogramming Procedure

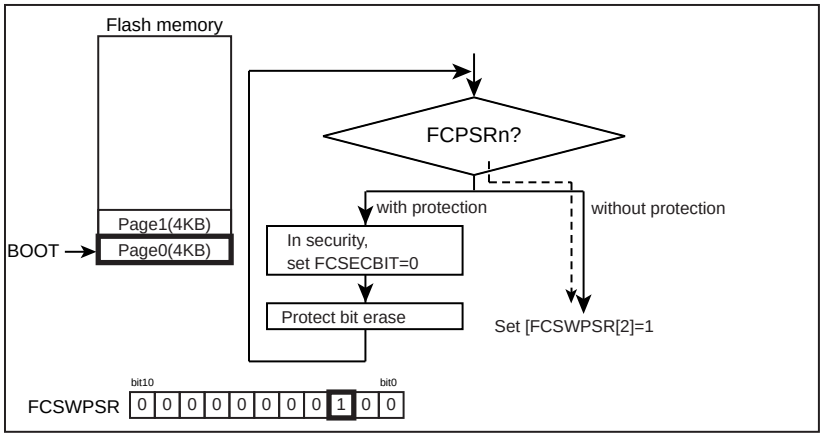
24.5.1.1 Step-1

Confirm if 0x0 is read from FCSWPSR[2:0].



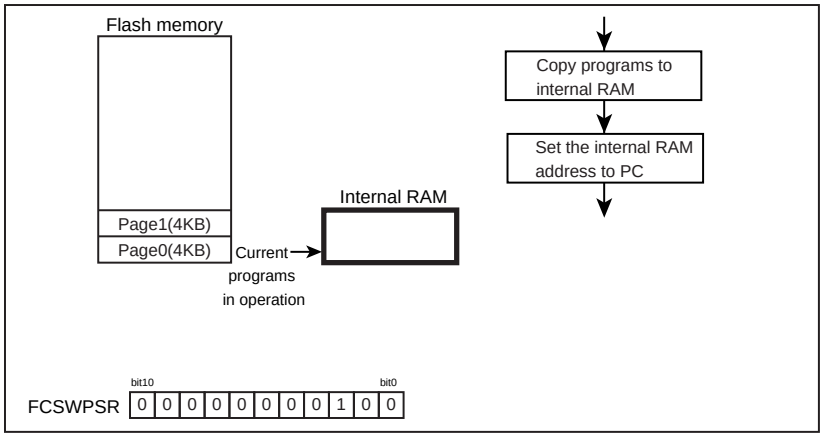
24.5.1.2 Step-2

Check each bit of FCPSR to confirm if the protect status is released. Then "1" is set to FCSWPSR[2] using automatic memory swap command.



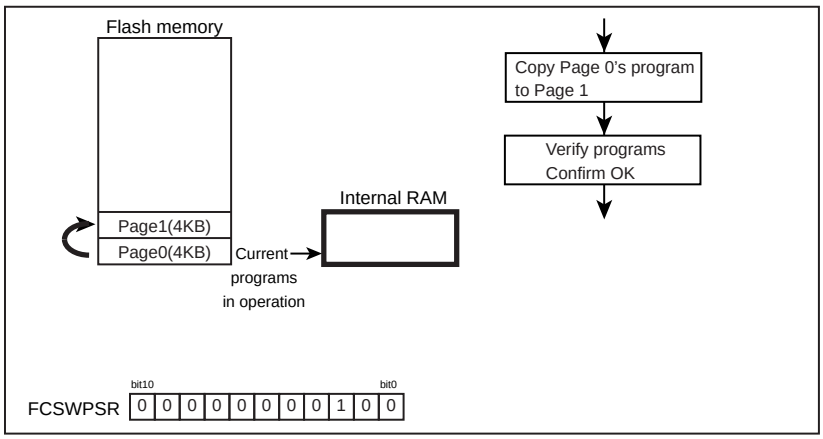
24.5.1.3 Step-3

Transfer the reprogramming routine to the internal RAM. Move PC (Program Counter) to the transferred program.



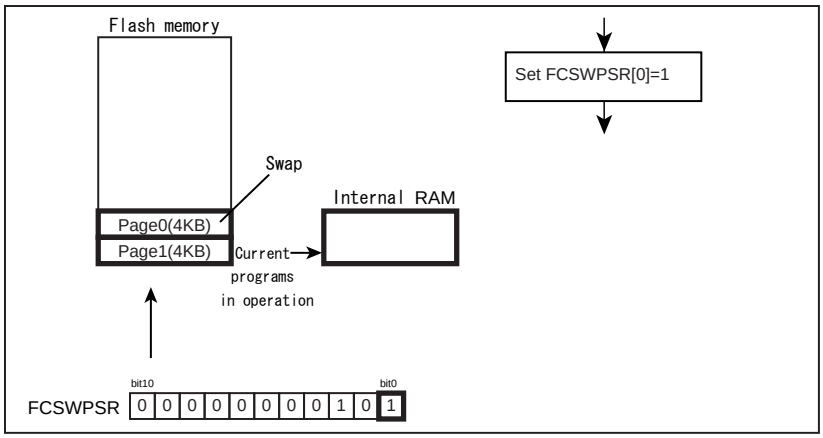
24.5.1.4 Step-4

Erase Page 1. Then write a program of Page0 to those of Page 1.



24.5.1.5 Step-5

Automatic memory swap command set "1" to FCSWPSR[0] to swap Page 0 with Page 1.

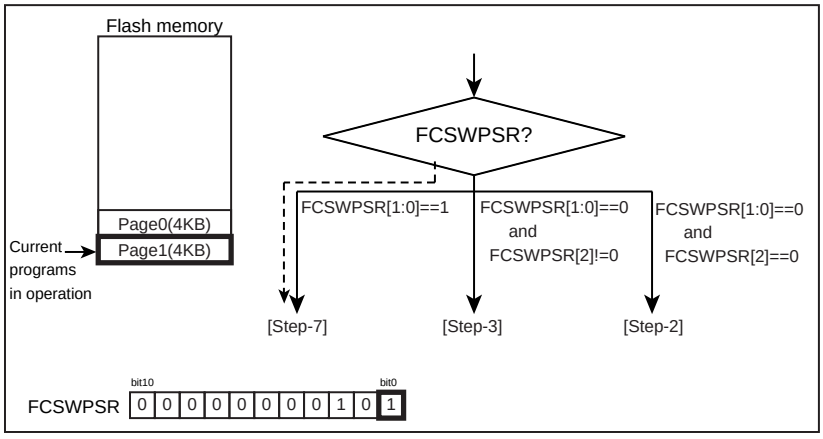


24.5.1.6 Step-6

Perform or release reset.

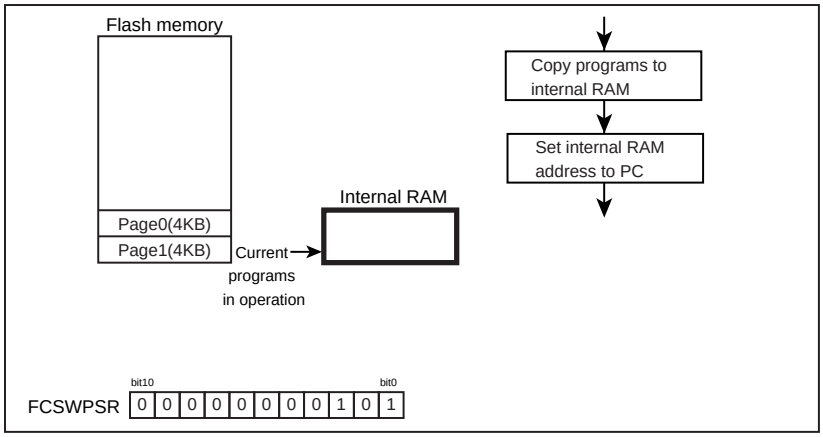
Page 1 is assigned to address 0 and Flash memory boots-up at Page 1.

A program branches to the conditioning routine that FCSWPSR[1:0] is set to "1". (To [Step-7])



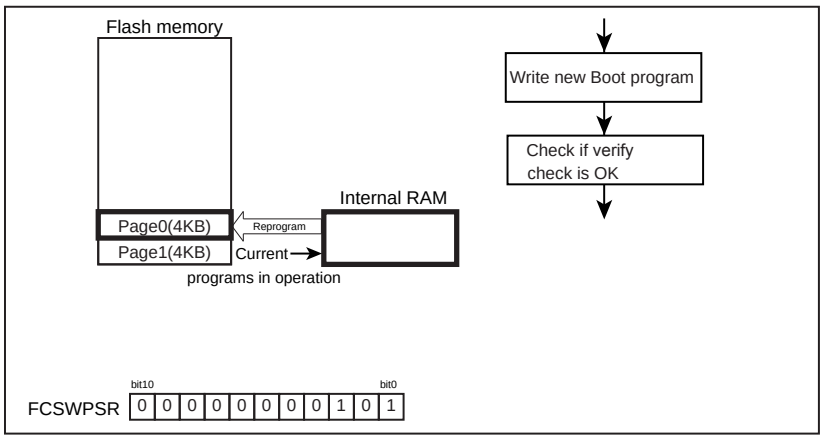
24.5.1.7 Step-7

Transfer the Flash reprogramming routine to the internal RAM then set the internal RAM address to PC (Program Counter).



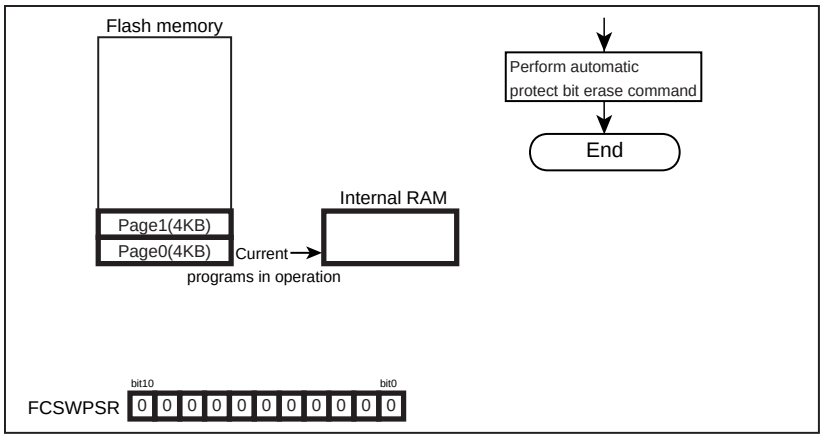
24.5.1.8 Step-8

Write a new boot program to Page0.



24.5.1.9 Step-9

Execute automatic protect bit erase command.



25. Debug Interface

25.1 Specification Overview

TMPM475FDFG/FZFG/FYFG contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the debugging tools and the Embedded Trace Macrocell™(ETM) unit for instruction trace output.Trace data is output to the dedicated pins(TRACEDATA[3:0], SWV) for the debugging via the on-chip Trace Port Interface Unit (TPIU).

For details about SWJ-DP, ETM and TPIU, refer to "ARM documentations set for the Cortex-M4F".

25.2 SWJ-DP

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

Pin name	Function	Description	I/O
TMS	JTAG	JTAG Test Mode Selection	Input
SWDIO	SW	Serial Wire Data Input/Output	I/O
TCK	JTAG	JTAG Test Clock	Input
SWCLK	SW	Serial Wire Clock	Input
TDO	JTAG	JTAG Test Data Output	Output
SWV	SW	(Serial Wire Viewer Output)	(Output) (Note)
TDI	JTAG	JTAG Test Data Input	Input
$\overline{\text{TRST}}$	JTAG	JTAG Test $\overline{\text{RESET}}$	Input

Note:When SWV function is used, this pin is used as output pin.

25.3 ETM

ETM supports four data signal pins (TRACEDATA[3:0]), one clock signal pin (TRACECLK) and trace output from Serial Wire Viewer (SWV) .

25.4 Peripheral Functions in Halt Mode

When the Cortex-M4F core enters in the halt mode, the watchdog-timer (WDT) automatically stops. It is selectable that 16-bit Timer(TMRB and TMR16A) continue or stop counting.Other peripheral functions continue to operate.

25.5 Connection with a Debug Tool

25.5.1 About connection with debug tool

Concerning a connection with debug tools, refer to manufactures recommendations.

Debug interface pins contain a pull-up resistor and a pull-down resistor.When debug interface pins are connected with external pull-up or pull-down, please pay attention to input level.

Note: Ensure that to measure the power-consumption with debug tool connected in STOP mode is prohibited.

25.5.2 Important points of using debug interface pins used as general-purpose ports

The debug interface pins can also be used as general-purpose ports.

After releasing reset, the particular pins of the debug interface pins are initialized as the debug interface pins. The other debug interface pins should be changed to the debug interface pins if needed.

If the debug interface pins are used as the general I/O port, please prepare the way to change the general I/O port to the debug interface pins beforehand.

Table 25-1 Example Table of using debug interface pins

	Debug interface pins						
	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA[3:0]	TRACE CLK
JTAG+SW (After reset)	o	o	o	o	o	×	×
JTAG+SW (without $\overline{\text{TRST}}$)	× (Note)	o	o	o	o	×	×
JTAG+TRACE	o	o	o	o	o	o	o
SW	×	×	×	o	o	×	×
SW+SWV	×	×	o	o	o	×	×
Debug disable	×	×	×	×	×	×	×

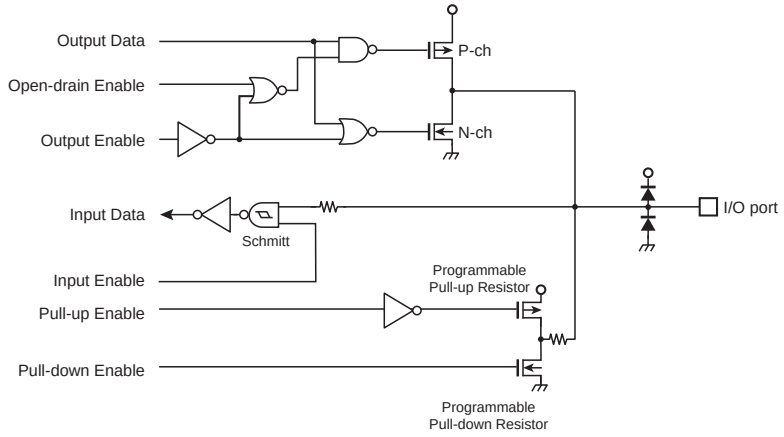
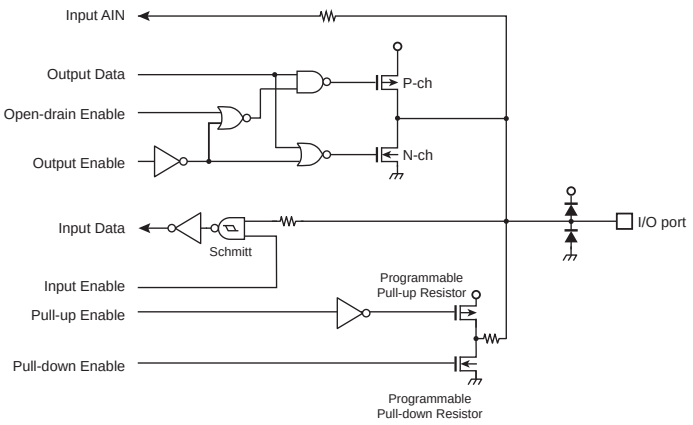
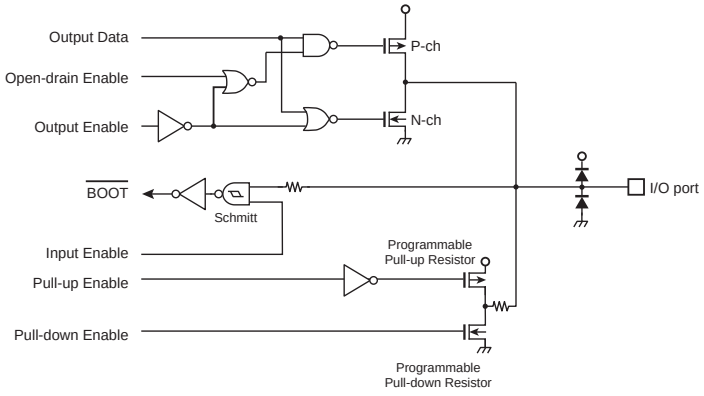
o : Enabled × : Disabled (Usable as general-purpose port)

Note: For the treatment of the pin of which the $\overline{\text{TRST}}$ function is assigned, select the $\overline{\text{TRST}}$ function with the function register and set the pin to OPEN or "High level".

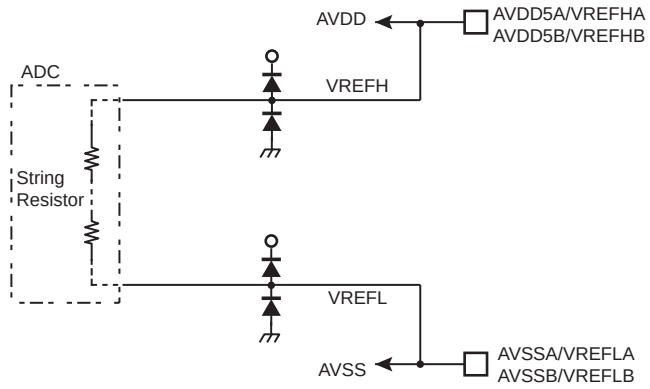
26. Port Section Equivalent Circuit Schematic

Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

26.1 Port

<table>
<tr><td>PA0 to 7, PB0 to 7, PC0 to 7,
PD0 to 6, PE0 to 7, PF1 to 4,
PG0 to 7,
PK0 to 1, PL0 to 1, PN0 to 2</td><td>(Programmable open drain output, Programmable Pull-up / Pull-down, Schmitt input)
 
</td></tr>
<tr><td>PH0 to 7, PJ0 to 7, PP0 to 3</td><td>(Analog input, Programmable open drain output, Programmable Pull-up / Pull-down, Schmitt input)
 
</td></tr>
<tr><td>PF0</td><td>(Boot input, Programmable open drain output, Programmable Pull-up / Pull-down, Schmitt input)
 
</td></tr>
</table>

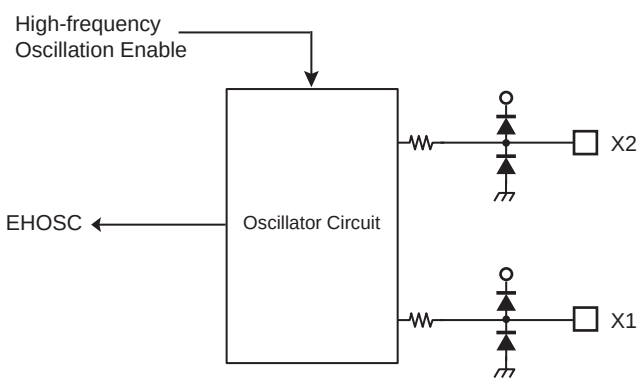
26.2 Analog terminal

VREFHA, VREFLA VREFHB, VREFLB	

26.3 Control terminal

MODE	 (Note)MODE pin is fixed to GND.
$\overline{\text{RESET}}$	

26.4 Clock terminal

X1, X2	

27. Electrical Characteristics

27.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5	−0.3 to 6	V
		RVDD5	−0.3 to 6	
		AVDD5A/B	−0.3 to 6	
Capacitor voltage		VOUT12	−0.3 to 3	V
		VOUT3	−0.3 to 3.9	
Input voltage		V_{IN}	−0.3 to VDD+0.3 (Note 2)	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	−5	
	Total	ΣI_{OH}	−50	
Power consumption	Ta = 85 °C	PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−55 to 125	°C
Operating Temperature		T_{OPR}	−40 to 85	°C

Note 1: Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

Note 2: VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B.

27.2 DC Electrical Characteristics (1/2)

DVSS = AVSSA = AVSSB = 0V, Ta = −40 to 85 °C

Parameter		Symbol	Rating		Min	Typ. (Note 1)	Max	Unit
Supply voltage (Note 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f_{OSC} = 8 to 10 MHz fsys = 1 to 120 MHz		4.5	−	5.5	V
Supply voltage (during Flash W/E) (Note 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f_{OSC} = 8 to 10 MHz fsys = 1 to 120 MHz (Ta = 0 to 70 °C)		4.5	−	5.5	V
Supply voltage (Power-on or Power-off) (Note 3)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f_{OSC} = 8 to 10 MHz fsys = 1 to 120 MHz		3.9	−	5.5	V
Low-level input voltage	Schmitt input	V_{IL1}	VDD = 4.5 to 5.5V (Note 4)		−0.3	−	0.25 VDD	V
High-level input voltage	Schmitt input	V_{IH1}	VDD = 4.5 to 5.5V (Note 4)		0.75 VDD	−	VDD+0.3	V
Capacitance (Note 5)		C_{out}	RVDD5 = 4.5 to 5.5V VOUT12, 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_{LI}	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		VTH	4.5 ≤ VDD ≤ 5.5 (Note 4)		0.3	0.6	−	V
Pin capacitance (Except power supply pins)		C_{IO}	fc = 1 MHz		−	−	10	pF

Note 1: Ta = 25 °C, DVDD5 = AVDD5A = AVDD5B = RVDD5 = 5V, unless otherwise noted.

Note 2: The same voltage must be supplied to DVDD5, AVDD5A, AVDD5B and RVDD5.

Note 3: It is a voltage range in the case of Power-on or Power-off (when VLTD disabled). In the range whose Power-line is 3.9V ≤ VDD < 4.5V, does not guarantee a 12-bit A/D converter and AC electrical Characteristics.

Note 4: VDD = DVDD5 = AVDD5A = AVDD5B = RVDD5

Note 5: VOUT12 and VOUT3 pin should be connected to GND via same value of capacitance. The IC outside cannot have the power supply from VOUT12 and VOUT3.

27.3 DC Electrical Characteristics (2/2)

DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5V to 5.5V, Ta = −40 to 85 °C

Parameter	Symbol	Condition		Min	Typ. (Note)	Max	Unit
		System clock (fsys)	Operating conditions				
NORMAL	I_{DD}	120MHz	Refer to Table 27-1 Table 27-2 regarding to the operation condition	−	45	100	mA
IDLE				−	13	50	mA
NORMAL		80MHz		−	34	75	mA
IDLE				−	9.8	38	mA
STOP		−		−	1.6	25	mA

Note 1: Ta = 25 °C, DVDD5 = AVDDA5 = AVDD5B = RVDD5 = 5V, unless otherwise noted.
Note 2: IDD at 80MHz are the reference values.

Table 27-1 I_{DD}Measurement Condition (Pin condition, Oscillator)

		NORMAL	IDLE	STOP
Pin condition	DVDD5 = AVDD5A = AVDD5B = RVDD5	4.5 to 5.5 V		
	X1, X2 pins	Connected to the oscillator		
	Input pin	Fixed		
	Output pin	Opened		
Operating conditions (Oscillator)	External high-speed oscillator (EHOSC)	Enabled		Disabled
	Internal high-speed oscillator (IHOSC)	Disabled		
	PLL for fsys	Enable (x12)		Disabled

Table 27-2 I_{DD}Measurement Condition (CPU, Peripheral circuit)

Circuit	The number of equipped circuits	NORMAL	IDLE	STOP
CPU	1	Enabled Dhrystone Ver2.1	Disabled	Disabled
A-VE	2	Enabled	Disabled	Disabled
µDMAC	1	Disabled	Disabled	Disabled
ADC	2	Enabled	Disabled	Disabled
TMRB	10	Enabled	Disabled	Disabled
WDT	1	Disabled	Disabled	Disabled
SIO/UART	4	Enabled	Disabled	Disabled
I2C	1	Disabled	Disabled	Disabled
PMD	2	Enabled	Disabled	Disabled
A-ENC	2	Enabled	Disabled	Disabled
I/O port	-	Enabled	Disabled	Disabled
VLTD	1	Enabled	Enabled	Disabled
OFD	1	Enabled	Disabled	Disabled
CAN	1	Disabled	Disabled	Disabled

27.4 12-bit ADC Electrical Characteristics

DVDD5 = RVDD5 = AVDD5A / VREFHA = AVDD5B / VREFHB = 4.5 V to 5.5 V
DVSS = AVSSA / VREFLA = AVSSB / VREFLB = 0V, Ta = −40 to 85 °C

Parameter	Symbol	Rating	Min	Typ.	Max	Unit
Analog reference voltage (+)	VREFHA VREFHB	−	−	AVDD	−	V
Analog input voltage	VAIN	−	AVSS	−	AVDD	V
Power supply current of analog reference voltage (Note 1)	IREF	−	−	3.5	5.0	mA
Supply current (Note 1) A/D conversion	−	Except for IREF	−	−	6.0	mA
INL error	−	AIN resistance ≤ 600 Ω AIN load capacitance ≥ 0.1 μF Conversion time ≥ 1.00 μs Conversion clock frequency = 120MHz (Note 4)	−	−	± 6	LSB
DNL error			−	−	± 5	
Offset error			−	−	± 5	
Full-scale error			−	−	± 6	
Total error			−	−	−10 to +6	

Note 1: The current is measured under the condition in which the only ADC is operating.
Note 2: 1LSB = (AVDD − AVSS) / 4096 [V]
Note 3: AVDD = AVDD5A =AVDD5B, AVSS = AVSSA = AVSSB
Note 4: The characteristic is measured under the condition in which 2 unit ADC is operating.

27.5 AC Electrical Characteristics

27.5.1 AC measurement condition

AC measurement condition

- Output levels: High = 0.8 × DVDD5 / Low = 0.2 × DVDD5
- Input levels: Refer to low-level input voltage and high-level input voltage in DC Electrical Characteristics.
- Load capacity:CL=30pF

Note: VDD = DVDD5 = AVDD5A = AVDD5B

27.5.2 Serial Channel Timing (SIO/UART)

27.5.2.1 I/O Interface mode (VDD = 4.5 to 5.5V)

In the table below, the letter x represents the period of the system clock (fsys). It varies depending on the programming of the clock gear function.

(1) SCLK input mode (Ta = -40 to 85°C)

[Data Input]

Parameter	Symbol	Equation		80 MHz		120 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	50	−	33.3	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	50	−	33.3	−	
SCLK cycle	t_{SCY}	8x	−	100	−	66.6	−	
Valid Data Input ← SCLK rise or fall (Note1)	t_{SRD}	30	−	30	−	30	−	
SCLK rise or fall → Input Data hold (Note 1)	t_{HSR}	x + 30	−	42.5	−	38.3	−	

[Data Output]

Parameter	Symbol	Equation		80 MHz		120M Hz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	4x	−	82.5 (Note3)	−	70.0 (Note3)	−	ns
SCLK Clock Low width (input)	t_{SCL}	4x	−	82.5 (Note3)	−	70.0 (Note3)	−	
SCLK cycle	t_{SCY}	8x	−	165	−	140	−	
Output Data ← SCLK rise or fall (Note 1)	t_{OSS}	$t_{SCY}/2 - 3x - 45$ (Note2)	−	0 (Note2)	−	0 (Note2)	−	
SCLK rise or fall → Output Data hold (Note 1)	t_{OHS}	$t_{SCY}/2$	−	82.5	−	70.0	−	

Note 1: SCLK rise or fall: In case of SCLK rise mode, the timing is SCLK rising. In case of SCLK fall mode, the timing is SCLK rising.

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. The value is not calculated.

(2) SCLK output mode (Ta = -40 to 85°C)

[Data Input / Output]

Parameter	Symbol	Equation		80 to 120 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable) (Note 3)	t_{SCY}	2x	−	50	−	ns
Output Data ← SCLK rise or fall	t_{OSS}	$t_{SCY}/2 - 25$ (Note1)	−	0 (Note2)	−	
SCLK rise or fall → Output Data hold	t_{OHS}	$t_{SCY}/2 - 25$ (Note1)	−	0 (Note2)	−	
Valid Data Input ← SCLK rise or fall	t_{SRD}	45	−	45	−	
SCLK rise or fall → Input Data hold	t_{HSR}	0	−	0	−	

Note 1: A calculated value should use it the SCLK cycle of the range which is not subtracted.
Note 2: The values show the minimum which are not subtracted. They are not calculated.
Note 3: Please adjust the SCLK cycle as tscy ≥ 50ns.

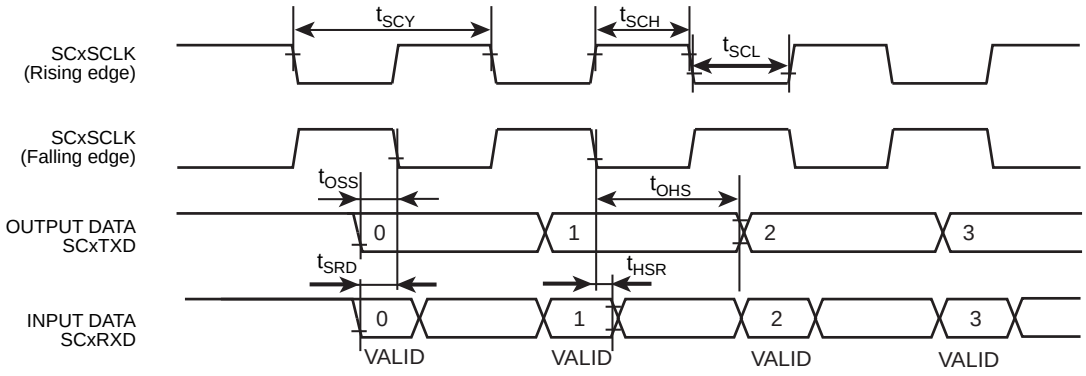


Figure 27-1 Serial channel timing(SIO)

27.5.3 Serial Bus Interface (I2C)

27.5.3.1 I2C Mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBIxCR.

Parameter	Symbol	Equation		Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	t_{SCL}	0	−	0	100	0	400	kHz
Hold time for START condition	$t_{HD;\ STA}$	−	−	4.0	−	0.6	−	μs
SCL Low width (Input) (Note 1)	t_{LOW}	−	−	4.7	−	1.3	−	μs
SCL High width (Input) (Note 2)	t_{HIGH}	−	−	4.0	−	0.6	−	μs
Setup time for a repeated START condition	$t_{SU;\ STA}$	(Note 5)	−	4.7	−	0.6	−	μs
Data hold time (Input) (Note 3) (Note 4)	$t_{HD;\ DAT}$	−	−	0.0	−	0.0	−	μs
Data setup time	$t_{SU;\ DAT}$	−	−	250	−	100	−	ns
Setup time for a STOP condition	$t_{SU;\ STO}$	−	−	4.0	−	0.6	−	μs
Bus free time between stop condition and start condition	t_{BUF}	(Note 5)	−	4.7	−	1.3	−	μs

Note 1: SCL clock Low width (output) = $(2^{n-1} + 58)/x$

Note 2: SCL clock High width (output) = $(2^{n-1} + 14)/x$

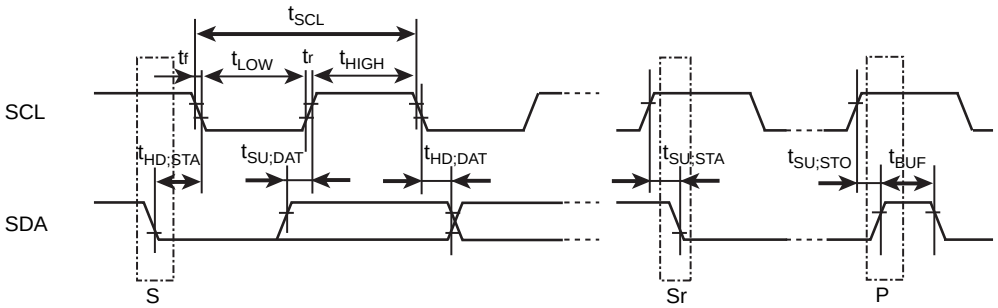
On I2C-bus specification, Maximum Speed of Standard Mode is 100kHz, Fast mode is 400kHz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

Note 3: The output data hold time is equal to 4x of internal SCL.

Note 4: The Philips I2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL. However, this SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software -dependent

Note 6: The Philips I2C-bus specification instructs that if the power supply to a Fast-mode device is switched off, the SDA and SCL I/O pins must be floating so that they don't obstruct the bus lines. However, this SBI does not satisfy this requirement.



S: Start condition
Sr: Re-start condition
P: Stop condition

Figure 27-2 Serial bus timing (I2C)

27.5.4 16-bit Timer / Event counter (TMRB)

27.5.4.1 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		120 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2x + 100	−	125	−	117	−	ns
Clock high pulse width	t_{VCKH}	2x + 100	−	125	−	117	−	ns

27.5.4.2 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		80 MHz		120 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	−	125	−	117	−	ns
High pulse width	t_{CPH}	2x + 100	−	125	−	117	−	ns

27.5.5 External Interrupt

In the table below, the letter x represents the fsys cycle time.

1. Except STOP release interrupts

Parameter	Symbol	Equation		80 MHz		120 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Low pulse width	t_{INTAL}	x + 100	−	112.5	−	108.3	−	ns
High pulse width	t_{INTAH}	x + 100	−	112.5	−	108.3	−	ns

2. STOP Release Interrupts

Parameter	Symbol	Equation		80 MHz		120 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Low pulse width	t_{INTAL}	100	−	100	−	100	−	ns
High pulse width	t_{INTAH}	100	−	100	−	100	−	ns

27.5.6 Debug Communication

27.5.6.1 AC measurement condition

- Output levels: High = 0.7 × DVDD5 / Low = 0.3 × DVDD5
- Load capacity : TRACECLK CL=25pF, TRACEDATA CL=20pF

27.5.6.2 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}	-	30	
DATA valid to CLK rising	T_{ds}	20	-	
DATA hold after CLK falling	T_{dh}	15	-	

27.5.6.3 JTAG Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	T_{dck}	100	-	ns
DATA hold after CLK rising	T_{d3}	4	-	
DATA valid after CLK rising	T_{d4}	-	50	
DATA valid to CLK rising	T_{ds}	20	-	
DATA hold after CLK falling	T_{dh}	15	-	

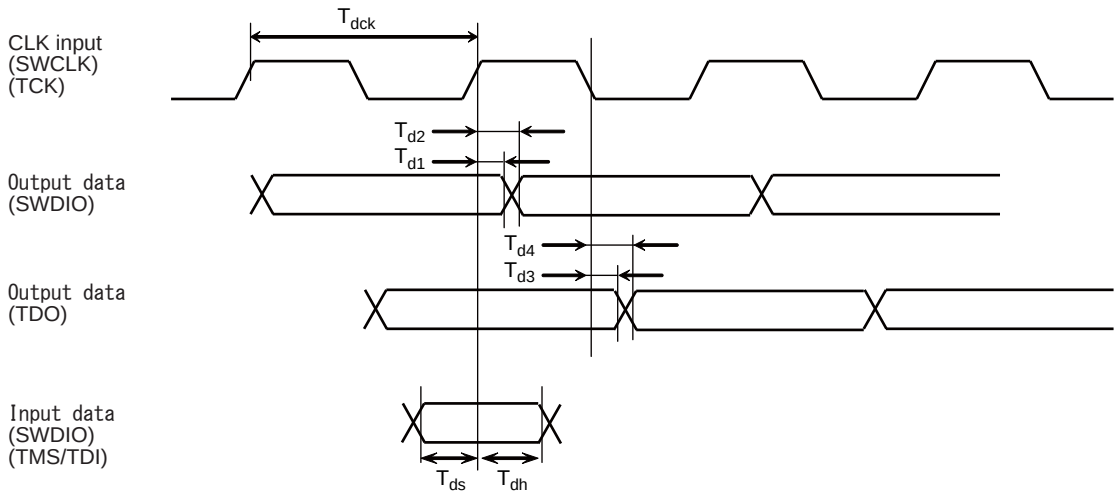


Figure 27-3 SWD / JTAG communication timing

27.5.7 EMT Trace

AC measurement conditions are as follows.

- Output levels: High = 0.7 × DVDD5 / Low = 0.3 × DVDD5
- Load capacity : TRACECLK CL=25pF, TRACEDATA CL=20pF

Parameter	Symbol	Min	Max	Unit
TRACECLK cycle	t_{tclk}	33.3	−	ns
TRACEDATA valid ← TRACECLK rise	t_{setupr}	2	−	
TRACECLK rise → TRACEDATA hold	t_{holdr}	1	−	
TRACEDATA valid ← TRACECLK fall	t_{setupf}	2	−	
TRACECLK fall → TRACEDATA hold	t_{holdf}	1	−	

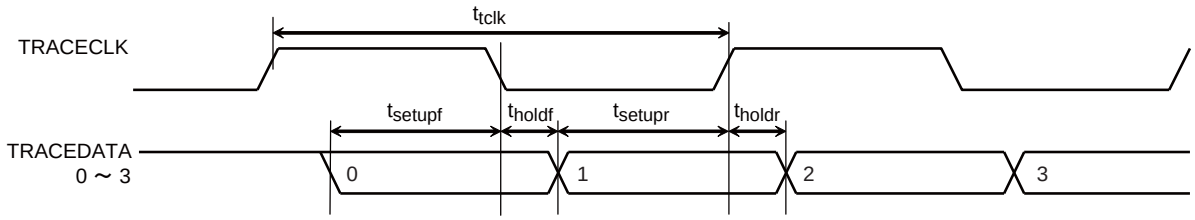


Figure 27-4 EMT trace timing

27.5.8 Flash Characteristics

Parameter	Rating	Min	Typ.	Max	Unit
Guarantee on Flash-memory rewriting	Ta = -40 to 85°C DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5 to 5.5	−	−	1000	times

27.5.9 On chip Oscillator

Parameter	Symbol	Rating	Min	Typ.	Max	Unit
Oscillation frequency (Note 1)	fiosc	Ta = -40 to 85°C	−	−	10	MHz
Accuracy of fiosc (Note1) (Note 2)	−	Ta = 0 to 50°C	−	−	± 1	%
		Ta = -40 to 0°C, Ta = 50 to 85°C	−	−	± 2	%

Note 1: This is the test value at factory shipment. If the described oscillation accuracy is required, the oscillation frequency must be trimmed not to exceed the maximum frequency of 120 MHz (the PLL setting is 12-fold) immediately after reflow soldering or later.

Note 2: This indicates errors of trimmed oscillation frequency at DVDD5 = RVDD5 = AVDD5A = AVDD5B =5.0V, and Ta=25°C.

27.5.10 External Oscillator

Parameter	Symbol	Rating	Min	Typ.	Max	Unit
High frequency Oscillation	feosc	Ta = -40 to 85°C	−	10	−	MHz

27.6 Oscillation Circuit

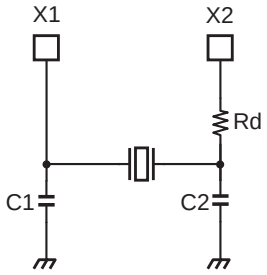


Figure 27-5 High-frequency oscillation connection

Note: To obtain a stable oscillation, load capacity and the position of the oscillator must be configured properly. Since these factors are strongly affected by substratepatterns, please evaluate oscillation stability using the substrate you use.

27.6.1 Ceramic Oscillator

The TMPM475FDFG/FZFG/FYFG recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.

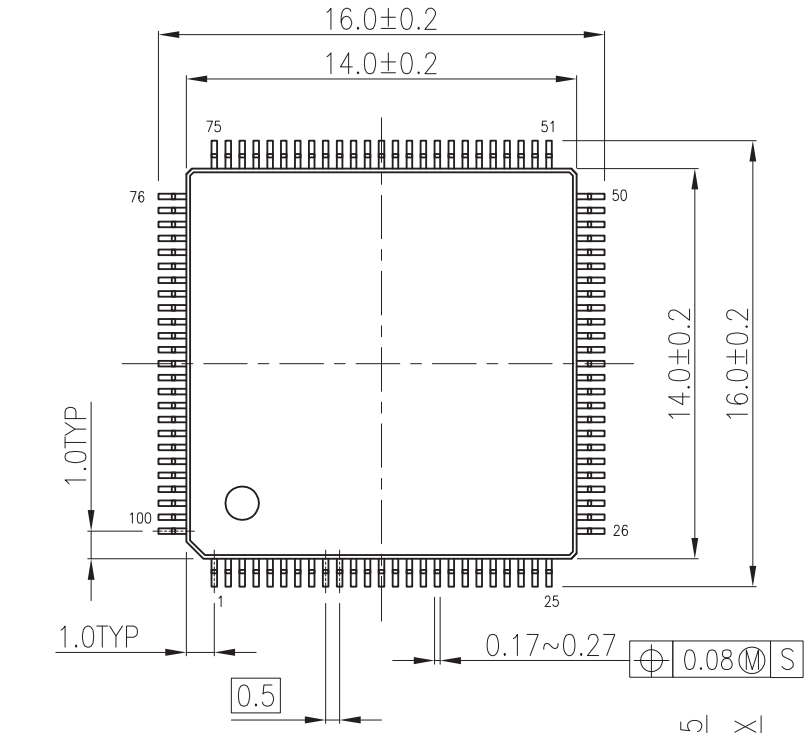
Please refer to the Murata Website for details.

28. Package Dimensions

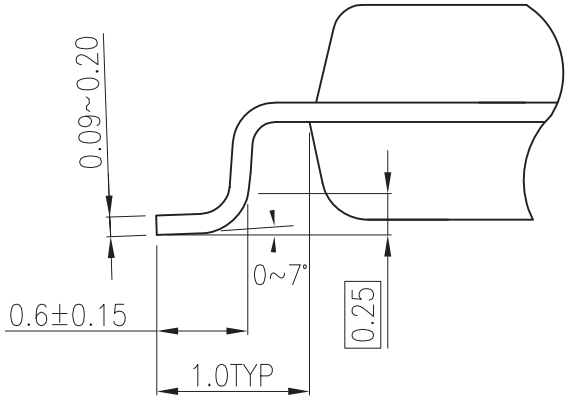
Type: P-LQFP100-1414-0.50-002

Unit: mm

Dimensions



Pin detail



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.**