TOSHIBA

32-bit TX System RISC

TX03 Series

TMPM380FYDFG
TMPM380FWDFG
TMPM380FYFG
TMPM380FWFG
TMPM382FWFG
TMPM382FSFG

TOSHIBA CORPORATION

Semiconductor Company

Revision History

Date	Revision	
2011/3/17	Rev 1	First Release

Comparison Table

	TMPM382FWFG TMPM382FSFG	TMPM380FYFG/DFG TMPM380FWFG/DFG
(1) Processor core	ARM Cortex-M3	
(2) Interrupt sources	- Internal: 55 factors - External: 8 factors INT0/1/2/3/4/5/8/F	- Internal: 77 factors - External: 16 factors INT0-F
(3) Input/Output ports	48 pins - Input/Output: 47 pins - Output: 1pin	84 pins - Input/Output: 83 pins - Output: 1pin
(4) Watchdog timer (WDT)	1 channel	
(5) Power-on Reset Circuit (POR)	1 channel	
(6) Voltage Detection Circuit (VLTD)	1 channel	
(7) Oscillation Frequency Detector (OFD)	1 channel	
(8) DMA contorller	2 channels	
(9) Encoder input circuit (ENC)	—	2 channels
(10) 16-bit Multi Purpose Timer (MPT)	1 channel ch0	3 channels ch0-2
(11) 16-bit timer (TMRB)	8 channels ch0-7 (ch3,5:16bit interval timer mode only)	8 channels ch0-7
(12) Real time clock (RTC)	1 channel	
(13) Serial channel (UART/SIO)	3 channels ch0/1/4	5 channels ch0-4
(14) Serial bus interface (I2C/SIO)	1 channel ch0	2 channels ch0-1
(15) Synchronous serial Port (SSP)	1 channel ch0	2 channels ch0-1
(16) Remote control signal preprocesser (RMC)	1 channel	
(17) 12-bit A/D converter (ADC)	1unit - 10 channels ch0-9	1unit - 18 channels ch0-17
(18) Standby mode	- Standby modes: IDLE, SLEEP and STOP - Sub clock operation (32.768KHz): SLOW, SLEEP	
(19) Clock generator(CG)	1 channel	
(20) Endian	Little endian	
(21) Maximum operating frequency	40MHz	
(22) Operating voltage range	4.0V~5.5V (with on-chip regulator)	
(23) Temperature range	-40°C~85°C (except during Flash writing/ erasing and debugging) 0°C~70°C (during Flash writing/ erasing and degugging)	

32-bit RISC microcontroller TX03 series

TMPM380FYFG, TMPM380FWFG TMPM380FYDFG, TMPM380FWDFG TMPM382FWFG, TMPM382FSFG

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

Product No.	On chip Flash ROM	On chip RAM	Package
TMPM380FYFG	256 Kbyte	16 Kbyte	LQFP100-P-1414-0.50H
TMPM380FYDFG ＊＊	256 Kbyte	16 Kbyte	QFP100-P-1420-0.65Q
TMPM380FWFG ＊＊	128 Kbyte	12 Kbyte	LQFP100-P-1414-0.50H
TMPM380FWDFG ＊＊	128 Kbyte	12 Kbyte	QFP100-P-1420-0.65Q
TMPM382FWFG ＊＊	128 Kbyte	12 Kbyte	QFP64-P-1414-0.80C
TMPM382FSFG ＊＊	64 Kbyte	8 Kbyte	QFP64-P-1414-0.80C

＊＊:Under development

1.1 Features

(1) ARM Cortex-M3 microcontroller core

1) Improved code efficiency has been realized through the use of Thumb®-2 instruction
   - New 16-bit Thumb® instructions for improved program flow
   - New 32-bit Thumb instructions for improved performance
   - Auto-switching between 32-bit instruction and 16-bit instruction is executed by compiler.

2) Both high performance and low power consumption have been achieved

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

■ Low power consumption
   - Optimized design using a low power consumption library
   - Standby function that stops the operation of the microcontroller core

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

(2) Interrupt sources
   - Internal: 77 factors…The order of precedence can be set over 7 levels (except NMI).
   - External: 16 factors…The order of precedence can be set over 7 levels

(3) Input/Output ports 84 pins
   - Input/Output: 83 pins Output: 1pin

(4) Watchdog timer (WDT): 1 channel
   - binary counter

(5) Power-on Reset Circuit (POR)

(6) Voltage Detection Circuit (VLTD)

(7) Oscillation Frequency Detector (OFD)

(8) DMA contorller: 2 channels
- Incr to Incr / Incr to No-Incr / No-Incr to Incr / No-Incr to No-Incr
- 4word FIFO buffer for each channel *2ch
- Scatter/gather transmission support

(9) Encoder input circuit (ENC): 2 channels
- Correspond to increamental encoder(AB/ABZ)
- Rotation direction detecttion
- Counter for absolute position detection
- Comparator for position detection
- Noise filter
- 3 phase sensor input

(10) 16-bit Multi Purpose Timer (MPT) : 3 channels

Channel 0/1: Supported for PMD fucntion / IGBT function / 16bit timer function
Channle 2: Supported for IGBT function / 16bit timer function

PMD function
- 3-phase PWM waveforms with the same PWM frequency
- Synctrigger signals to the AD converter
- The protection circuit controls in emergency

IGBT function
- 16-bit PPG output (Two output pins)
- External-triggered start and stop
- The protection circuit controls in emergency

16bit timer function
- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG output (One output pin)
- Input capture function

(11) 16-bit timer (TMRB): 8 channels
- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG output
- External trigger programmable square-wave output mode (PPG)
- Timer synchronous mode
- Input capture function

(12) Real time clock (RTC)
- Clock (hour, minute and second)
- Calender (Month, week, date and leap year)
- Alarm (Alarm output)
- Alarm interruption

(13) Serial channel (UART/SIO): 5 channels
- Either UART mode or synchronous mode can be selected (4byte FIFO equipped)

(14) Serial bus interface (I2C/SIO): 2 channels
- Either I2C bus mode or synchronous mode can be selected

(15) Synchronous serial Port (SSP): 2 channels
- SPI flame format /SSI flame format /Microwire flame format
- 16byte FIFO equipped (16bit*8)

(16) Remote control signal preprocesser (RMC) : 1 channel
- Can receive up to 72bit data at a time

(17) 12-bit A/D converter (ADC): 1unit (18 channels for analog input)
- Start by the internal trigger: TMRB interrupt / PMD trigger
- 3 conversion mode(Trigger start,Software start,Constant conversion) Arbitrary AIN can be selected
- AD Conversion Result Register (12ch)
- AD conversion monitoring function (2ch)
- Conversion speed 2.0usec (@ ADC conversion clock = 40MHz)

(18) Standby mode
- Standby modes: IDLE, SLEEP and STOP
- Sub clock operation (32.768KHz): SLOW, SLEEP

(19) Clock generator(CG)
- External Osciillator (High Freq. 10MHz X'tal/Ceramic) or On-chip Oscillator (9MHz)
- External Oscillator (Low Freq. 32KHz X'tal)
- On-chip PLL (4 times)
- Clock gear function: The high-speed clock can be divided into 1/1, 1/2, 1/4, 1/8, 1/16

(20) Endian
- Little endian

(21) Maximum operating frequency
- 40MHz

(22) Operating voltage range
- 4.0V~5.5V (with on-chip regulator)

(23) Temperature range
- -40°C~85°C (except during Flash writing/ erasing and debugging)
- 0°C~70°C (during Flash writing/ erasing and degugging)

1.2 Block Diagram

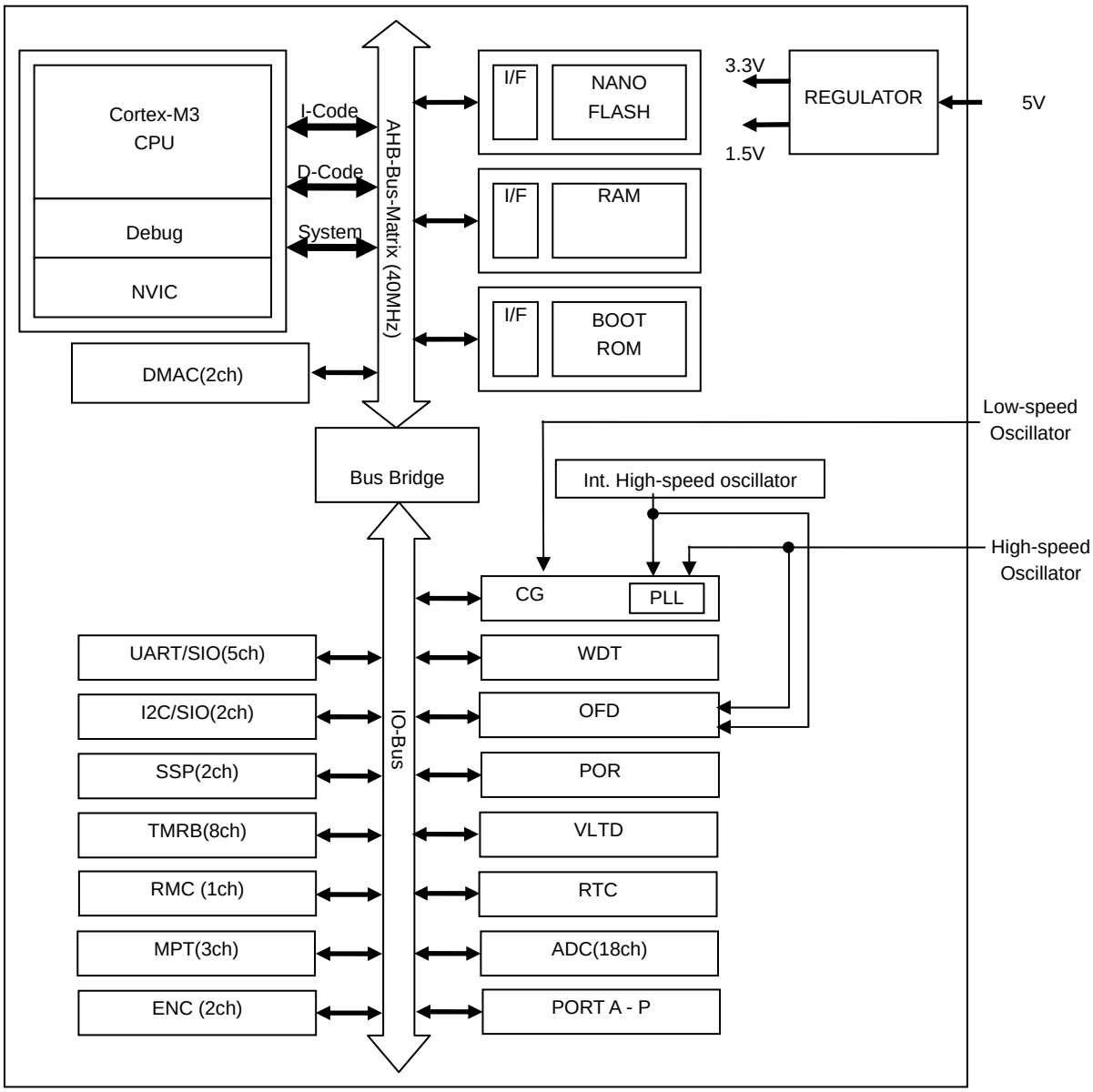


Fig1-1 TMPM380 block diagram

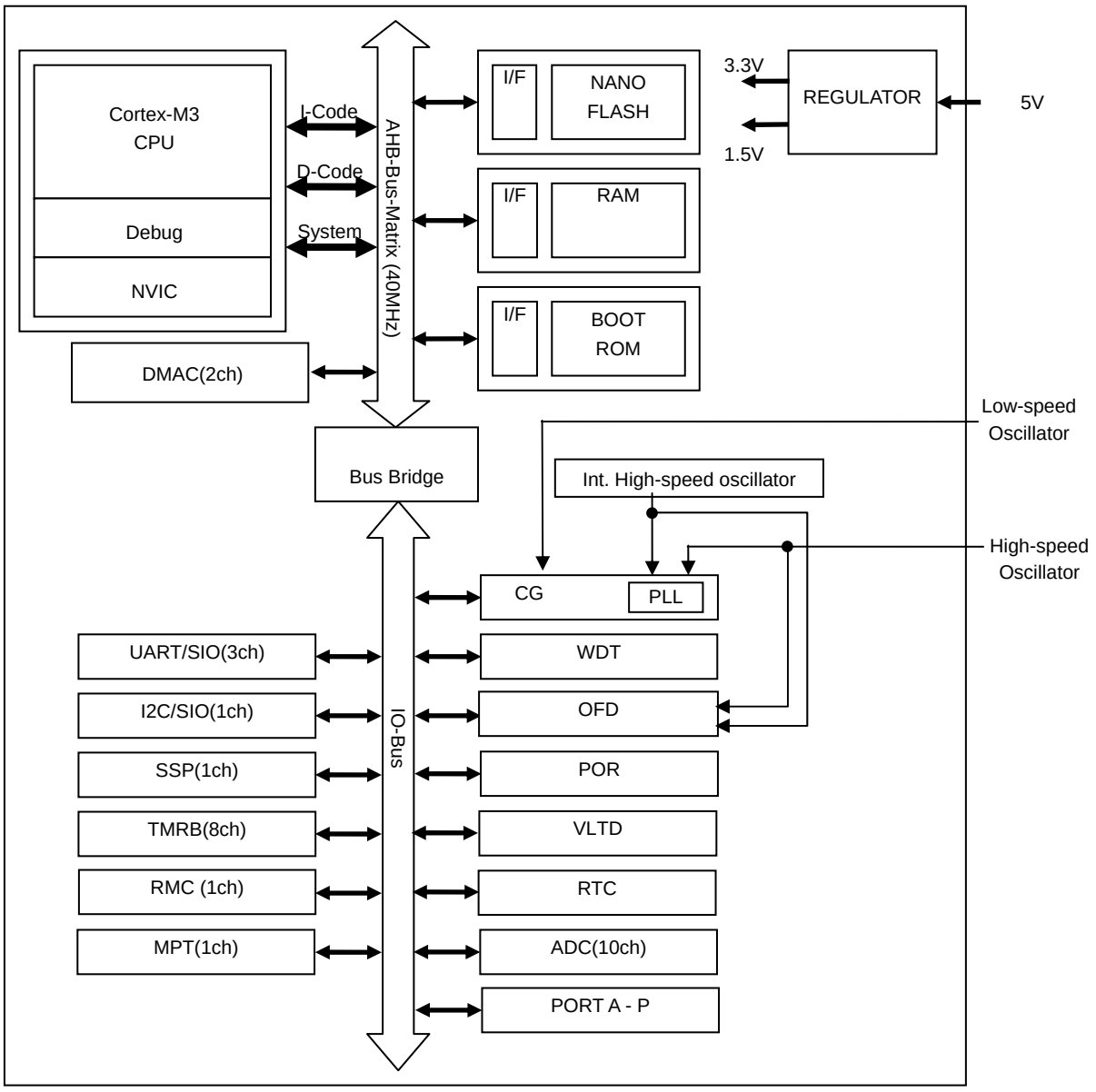


Fig1-1 TMPM382 block diagram

2 Pin Layout and Pin Functions

This chapter describes the pin layout, pin names and pin functions of TMPM380FYFG, TMPM380FWFG, TMPM380FYDFG,TMPM380FWDFG TMPM382FWFG and TMPM382FSFG.

2.1 Pin Layout (Top view)

Fig.2-1 shows the pin layout of TMPM380FYFG and TMPM380FWFG.

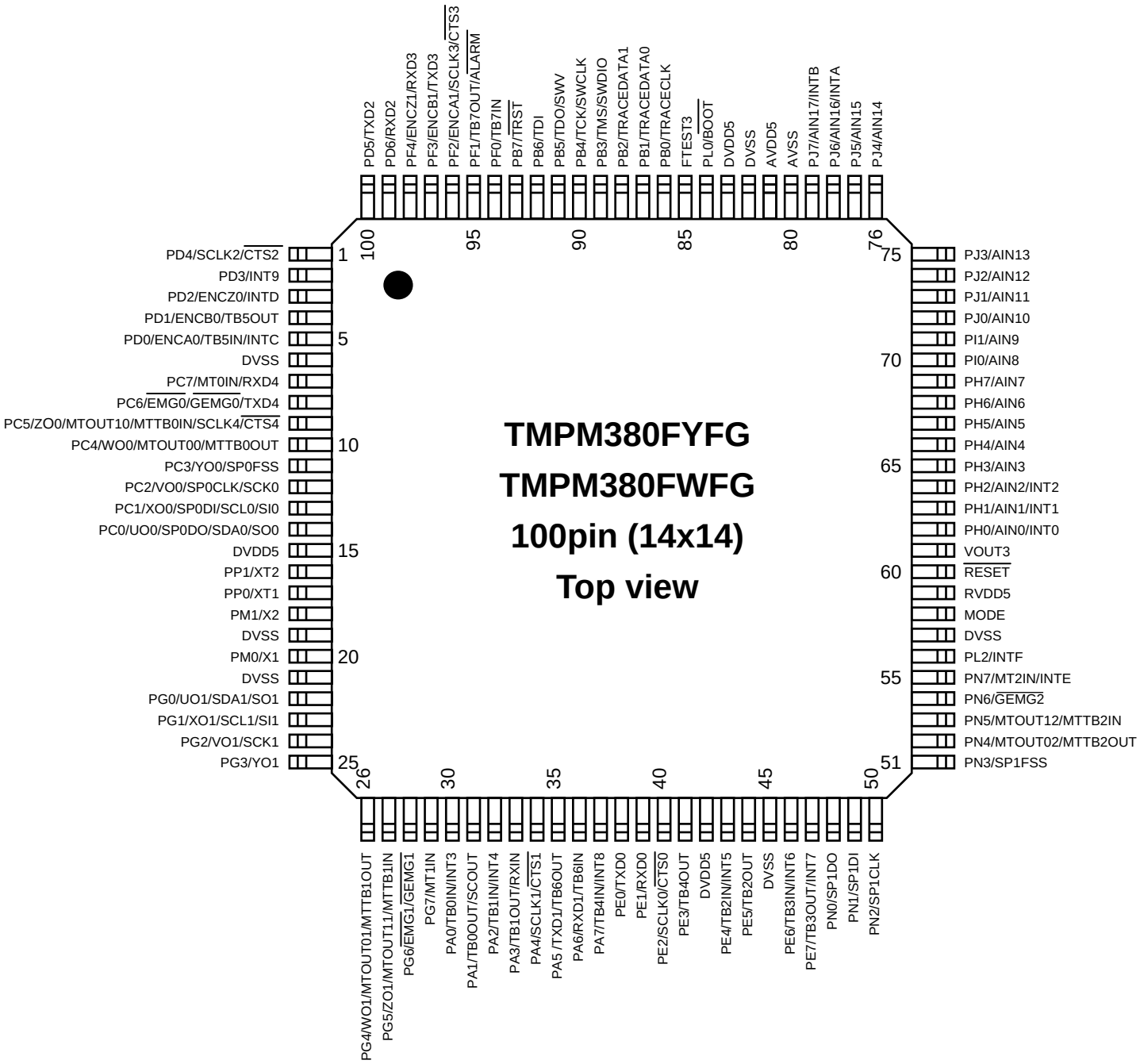


Fig. 2-1 Pin layout (TMPM380FxFG)

Fig.2-2 shows the pin layout of TMPM380FYDFG and TMPM380FWDFG.

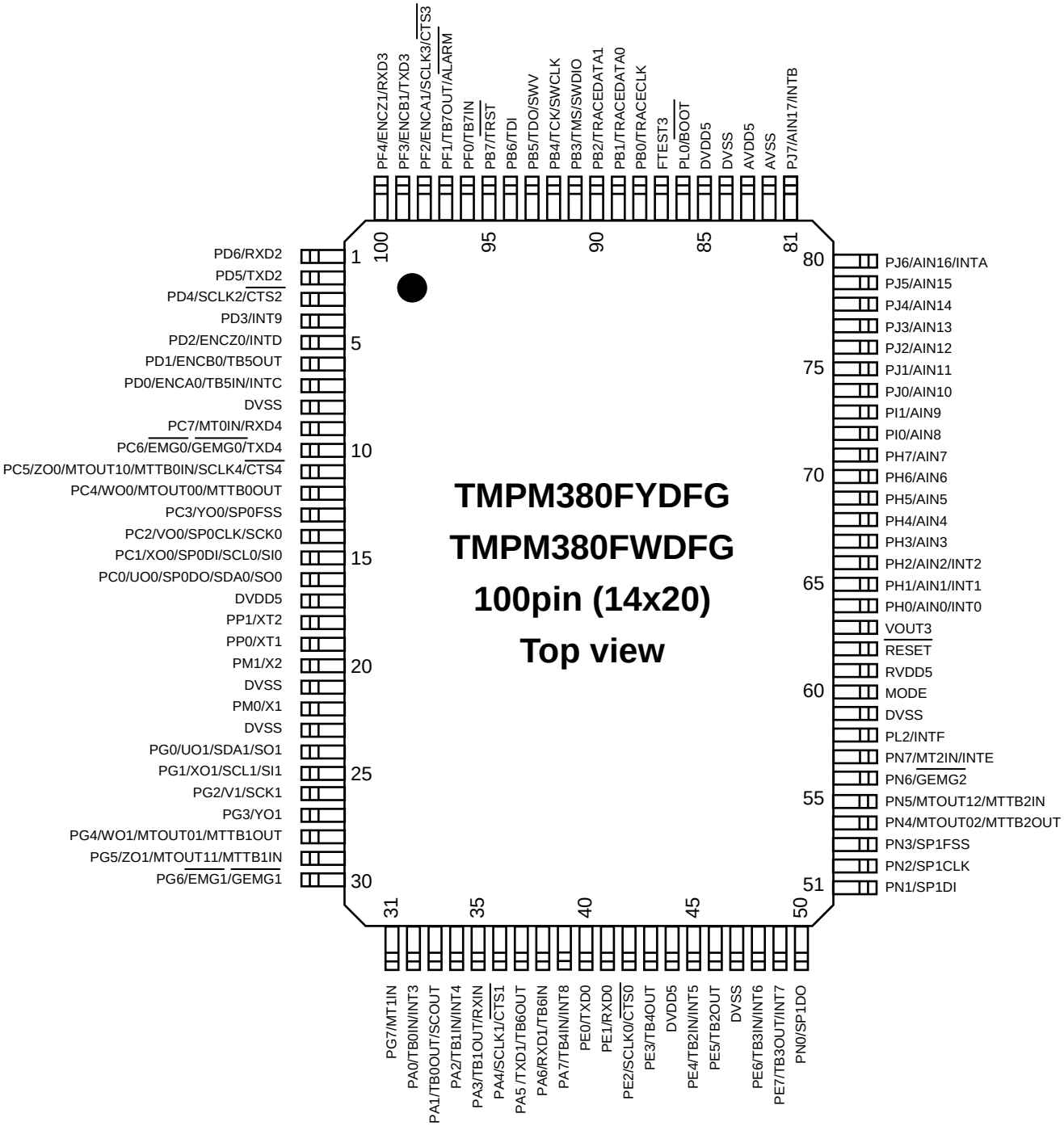


Fig. 2-2 Pin layout (TMPM380FxDFG)

Fig.2-3 shows the pin layout of TMPM382FWFG and TMPM382FSFG.

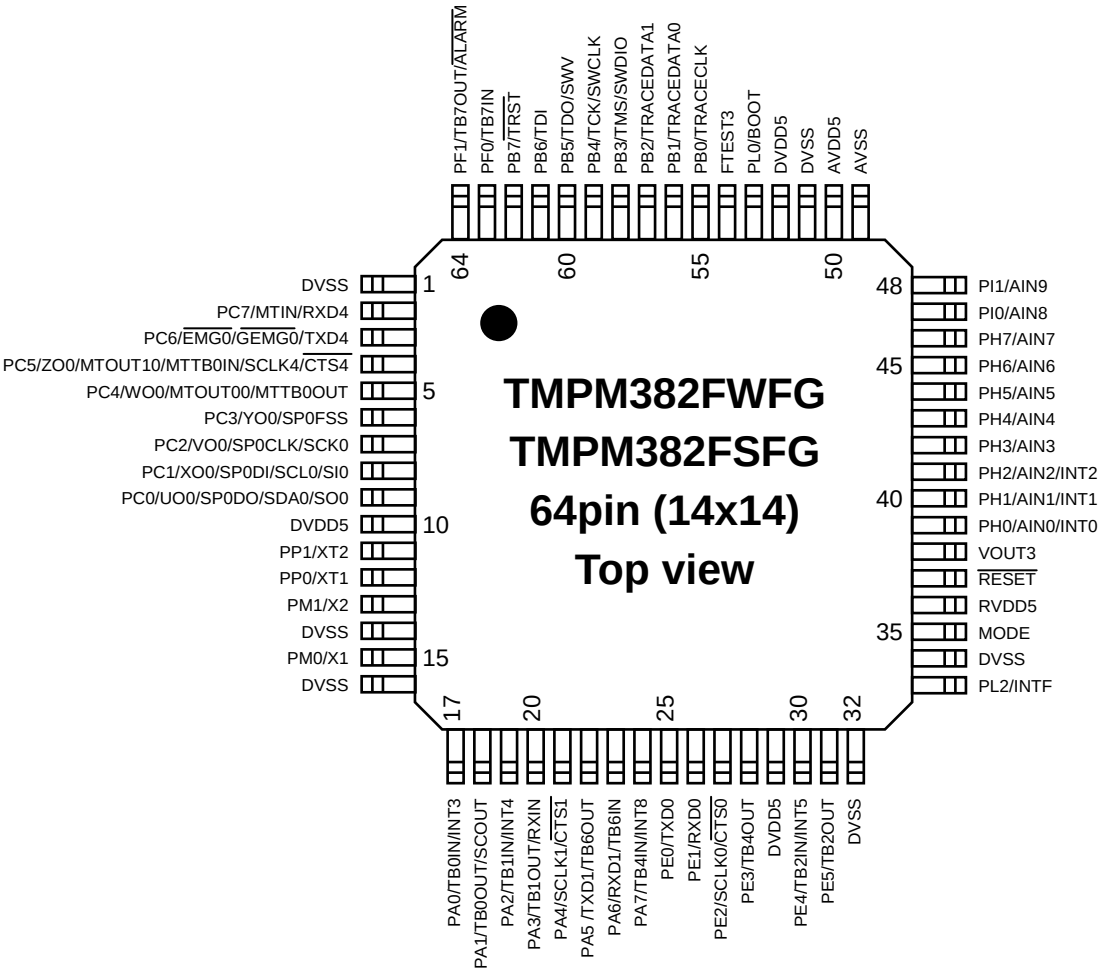


Fig. 2-3 Pin layout (TMPM382FxFG)

2.2 Pin function

Table 2-1 lists the pin functions of TMPM380FxFG/DFG. Table 2-4 shows the operating voltage of each pin, and Table 2-5 shows the voltage range of every pin.

Table 2-1 Pin functions (1/5)

M380FxDFG QFP100	M380FxFG LQFP100	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
3	1	PD4 SCLK2 $\overline{\text{CTS2}}$	Hi-Z	O	O
4	2	PD3 INT9	Hi-Z	O	O
5	3	PD2 ENCZ0 INTD	Hi-Z	O	O
6	4	PD1 ENCB0 TB5OUT	Hi-Z	O	O
7	5	PD0 ENCA0 TB5IN INTC	Hi-Z	O	O
8	6	DVSS			
9	7	PC7 MT0IN RXD4	Hi-Z	O	O
10	8	PC6 $\overline{\text{EMG0}}$ $\overline{\text{GEMG0}}$ TXD4	Hi-Z	O	O
11	9	PC5 ZO0 MTOUT10 MTTB0IN SCLK4 $\overline{\text{CTS4}}$	Hi-Z	O	O
12	10	PC4 WO0 MTOUT00 MTTB0OUT	Hi-Z	O	O
13	11	PC3 YO0 SP0FSS	Hi-Z	O	O
14	12	PC2 VO0 SP0CLK SCK0	Hi-Z	O	O
15	13	PC1 XO0 SP0DI SCL0 / SI0	Hi-Z	O	O

Table 2-1 Pin functions (2/5)

M380FxDFG QFP100	M380FxFG LQFP100	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
16	14	PC0 UO0 SP0DO SDA0 / SO0	Hi-Z	O	O
17	15	DVDD5			
18	16	PP1 XT2	Hi-Z	O	O
19	17	PP0 XT1	Hi-Z	O	O
20	18	PM1 X2	Hi-Z	O	O
21	19	DVSS			
22	20	PM0 X1	Hi-Z	O	O
23	21	DVSS			
24	22	PG0 UO1 SDA1 / SO1	Hi-Z	O	O
25	23	PG1 XO1 SCL1 / SI1	Hi-Z	O	O
26	24	PG2 VO1 SCK1	Hi-Z	O	O
27	25	PG3 YO1	Hi-Z	O	O
28	26	PG4 WO1 MTOUT01 MTTB1OUT	Hi-Z	O	O
29	27	PG5 ZO1 MTOUT11 MTTB1IN	Hi-Z	O	O
30	28	PG6 $\overline{\text{EMG1}}$ $\overline{\text{GEMG1}}$	Hi-Z	O	O
31	29	PG7 MT1IN	Hi-Z	O	O
32	30	PA0 TB0IN INT3	Hi-Z	O	O
33	31	PA1 TB0OUT SCOUT	Hi-Z	O	O
34	32	PA2 TB1IN INT4	Hi-Z	O	O
35	33	PA3 TB1OUT RXIN	Hi-Z	O	O

Table 2-1 Pin functions (3/5)

M380FxDFG QFP100	M380FxFG LQFP100	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
36	34	PA4 SCLK1 $\overline{\text{CTS1}}$	Hi-Z	O	O
37	35	PA5 TXD1 TB6OUT	Hi-Z	O	O
38	36	PA6 RXD1 TB6IN	Hi-Z	O	O
39	37	PA7 INT8 TB4IN	Hi-Z	O	O
40	38	PE0 TXD0	Hi-Z	O	O
41	39	PE1 RXD0	Hi-Z	O	O
42	40	PE2 SCLK0 $\overline{\text{CTS0}}$	Hi-Z	O	O
43	41	PE3 TB4OUT	Hi-Z	O	O
44	42	DVDD5			
45	43	PE4 TB2IN INT5	Hi-Z	O	O
46	44	PE5 TB2OUT	Hi-Z	O	O
46	45	DVSS			
48	46	PE6 TB3IN INT6	Hi-Z	O	O
49	46	PE7 TB3OUT INT7	Hi-Z	O	O
50	48	PN0 SP1DO	Hi-Z	O	O
51	49	PN1 SP1DI	Hi-Z	O	O
52	50	PN2 SP1CLK	Hi-Z	O	O
53	51	PN3 SP1FSS	Hi-Z	O	O
54	52	PN4 MTOUT02 MTTB2OUT	Hi-Z	O	O
55	53	PN5 MTOUT12 MTTB2IN	Hi-Z	O	O
56	54	PN6 $\overline{\text{GEMG2}}$	Hi-Z	O	O

Table 2-1 Pin functions (4/5)

M380FxDFG QFP100	M380FxFG LQFP100	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
57	55	PN7 MT2IN INTE	Hi-Z	O	O
58	56	PL2 INTF	Hi-Z	O	O
59	57	DVSS			
60	58	MODE	Hi-Z	O	
61	59	RVDD5			
62	60	$\overline{\text{RESET}}$	Pull UP	O	
63	61	VOUT3	VOUT3		
64	62	PH0 AIN0 INT0	Hi-Z	O	O
65	63	PH1 AIN1 INT1	Hi-Z	O	O
66	64	PH2 AIN2 INT2	Hi-Z	O	O
67	65	PH3 AIN3	Hi-Z	O	O
68	66	PH4 AIN4	Hi-Z	O	O
69	67	PH5 AIN5	Hi-Z	O	O
70	68	PH6 AIN6	Hi-Z	O	O
71	69	PH7 AIN7	Hi-Z	O	O
72	70	PI0 AIN8	Hi-Z	O	O
73	71	PI1 AIN9	Hi-Z	O	O
74	72	PJ0 AIN10	Hi-Z	O	O
75	73	PJ1 AIN11	Hi-Z	O	O
76	74	PJ2 AIN12	Hi-Z	O	O
77	75	PJ3 AIN13	Hi-Z	O	O
78	76	PJ4 AIN14	Hi-Z	O	O
79	77	PJ5 AIN15	Hi-Z	O	O
80	78	PJ6 AIN16 INTA	Hi-Z	O	O
81	79	PJ7 AIN17 INTB	Hi-Z	O	O

Table 2-1 Pin function (5/5)

M380FxDFG QFP100	M380FxFG LQFP100	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
82	80	AVSS			
83	81	AVDD5			
84	82	DVSS			
85	83	DVDD5			
86	84	PL0 $\overline{\text{BOOT}}$	Pull Up		O
87	85	FTEST3	Hi-Z(Note)		
88	86	PB0 TRACECLK	Hi-Z	O	O
90	87	PB1 TRACEDATA0	Hi-Z	O	O
90	88	PB2 TRACEDATA1	Hi-Z	O	O
91	90	PB3 TMS SWDIO	Pull Up	O	O
92	90	PB4 TCK SWCLK	Pull Down	O	O
93	91	PB5 TDO SWV	Hi-Z	O	O
94	92	PB6 TDI	Pull Up	O	O
95	93	PB7 $\overline{\text{TRST}}$	Pull Up	O	O
96	94	PF0 TB7IN	Hi-Z	O	O
97	95	PF1 TB7OUT $\overline{\text{ALARM}}$	Hi-Z	O	O
98	96	PF2 ENCA1 SCLK3 $\overline{\text{CTS3}}$	Hi-Z	O	O
99	97	PF3 ENCB1 TXD3	Hi-Z	O	O
100	98	PF4 ENCZ1 RXD3	Hi-Z	O	O
1	99	PD6 RXD2	Hi-Z	O	O
2	100	PD5 TXD2	Hi-Z	O	O

(Note) OPEN : Don't connect any circuit. This pin use for internal test only.

Table 2-2 lists the pin functions of TMPM382FxFG. Table 2-4 shows the operating voltage of each pin, and Table 2-5 shows the voltage range of every pin.

Table 2-2 Pin functions (1/4)

Pin No.	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
1	DVSS			
2	PC7 MTIN RXD4	Hi-Z	O	O
3	PC6 $\overline{\text{EMG0}}$ $\overline{\text{GEMG0}}$ TXD4	Hi-Z	O	O
4	PC5 ZO0 MTOUT10 MTTB0IN SCLK4 $\overline{\text{CTS4}}$	Hi-Z	O	O
5	PC4 WO0 MTOUT00 MTTB0OUT	Hi-Z	O	O
6	PC3 YO0 SP0FSS	Hi-Z	O	O
7	PC2 VO0 SP0CLK SCK0	Hi-Z	O	O
8	PC1 XO0 SP0DI SCL0 / SI0	Hi-Z	O	O
9	PC0 UO0 SP0DO SDA0 / SO0	Hi-Z	O	O
10	DVDD5			
11	PP1 XT2	Hi-Z	O	O
12	PP0 XT1	Hi-Z	O	O
13	PM1 X2	Hi-Z	O	O
14	DVSS			
15	PM0 X1	Hi-Z	O	O
16	DVSS			

Table 2-3 Pin functions (2/4)

Pin No.	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
17	PA0 TB0IN INT3	Hi-Z	O	O
18	PA1 TB0OUT SCOUT	Hi-Z	O	O
19	PA2 TB1IN INT4	Hi-Z	O	O
20	PA3 TB1OUT RXIN	Hi-Z	O	O
21	PA4 SCLK1 $\overline{\text{CTS1}}$	Hi-Z	O	O
22	PA5 TXD1 TB6OUT	Hi-Z	O	O
23	PA6 RXD1 TB6IN	Hi-Z	O	O
24	PA7 INT8 TB4IN	Hi-Z	O	O
25	PE0 TXD0	Hi-Z	O	O
26	PE1 RXD0	Hi-Z	O	O
27	PE2 SCLK0 $\overline{\text{CTS0}}$	Hi-Z	O	O
28	PE3 TB4OUT	Hi-Z	O	O
29	DVDD5			
30	PE4 TB2IN INT5	Hi-Z	O	O
31	PE5 TB2OUT	Hi-Z	O	O
32	DVSS			

Table 2-2 Pin functions (3/4)

Pin No.	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
33	PL2 INTF	Hi-Z	O	O
34	DVSS			
35	MODE	Hi-Z	O	
36	RVDD5			
37	$\overline{\text{RESET}}$	Pull UP	O	
38	VOUT3	VOUT3		
39	PH0 AIN0 INT0	Hi-Z	O	O
40	PH1 AIN1 INT1	Hi-Z	O	O
41	PH2 AIN2 INT2	Hi-Z	O	O
42	PH3 AIN3	Hi-Z	O	O
43	PH4 AIN4	Hi-Z	O	O
44	PH5 AIN5	Hi-Z	O	O
45	PH6 AIN6	Hi-Z	O	O
46	PH7 AIN7	Hi-Z	O	O
47	PI0 AIN8	Hi-Z	O	O
48	PI1 AIN9	Hi-Z	O	O

Table 2-2 Pin function (4/4)

Pin No.	Pin name	Output during Reset	SCHIMITT (O:Yes)	Open Drain mode
49	AVSS			
50	AVDD5			
51	DVSS			
52	DVDD5			
53	PL0 $\overline{\text{BOOT}}$	Pull Up		O
54	FTEST3	Hi-Z(Note)		
55	PB0 TRACECLK	Hi-Z	O	O
56	PB1 TRACEDATA0	Hi-Z	O	O
57	PB2 TRACEDATA1	Hi-Z	O	O
58	PB3 TMS SWDIO	Pull Up	O	O
59	PB4 TCK SWCLK	Pull Down	O	O
60	PB5 TDO SWV	Hi-Z	O	O
61	PB6 TDI	Pull Up	O	O
62	PB7 $\overline{\text{TRST}}$	Pull Up	O	O
63	PF0 TB7IN	Hi-Z	O	O
64	PF1 TB7OUT $\overline{\text{ALARM}}$	Hi-Z	O	O

(Note) OPEN : Don’t connect any circuit. This pin use for internal test only.

Table 2-4 Operating voltage of each Pin

Pin name	Operating voltage	Function
PM0,PM1 X1,X2	DVDD5 Internal 1.5V	X1 can't be driven by outside oscillator.
PP0,PP1 XT1,XT2	DVDD5 Internal 1.5V	XT1 can't be driven by outside oscillator.
$\overline{\text{RESET}}$	DVDD5	
MODE		Must be connected to GND
PA to PG,PL,PN I/O		
PH,PI,PJ AIN0 to AIN17	AVDD5(VREFH)	

Table 2-5 voltage range of each pin

Pin name	Voltage range	Function
RVDD5	4.0～5.5V	For internal circuit
DVDD5		For I/O ports
AVDD5		For ADC
VOUT3	2.7～3.6V	Output terminal of internal power supply. Connected to DVSS through 1 to 4.7μF capacitor
DVSS	GND	
AVSS		

Note : VOUT3 can't supply the power supply to an external circuit.

3 Processor Core

3.1 Processor Core

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

The following table shows the revision of the processor core in the TMPM380/M382. For further information on each revision, see the documents issued by ARM Limited.

Product Name	Core Revision
TMPM380	r2p0-00rel0
TMPM382	r2p0-00rel0

3.2 Configurable Opions

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

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

O: Implement

X: Not implement

3.3 Event

TMPM380/M382 does not support event input/output. Do not use SEV instruction and WFE instruction.

3.4 SLEEPDEEPl

TMPM380/M382 does not support SLEEPDEEP.

3.5 Exclusive access

TMPM380/M382 does not support EXCLUSIVE ACCESS.

3.6 Reset operation

3.6.1 Initial state

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

3.6.2 Reset operation

As the precondition, ensure that an internal high-frequency oscillator provides stable oscillation while power supply voltage is in the operating range. To reset the TMPM380/M382, input $\overline{\text{RESET}}$ signal at low level for a minimum duration of 12 system clocks (1.2μs with external 10MHz oscillator).

3.6.3 After Reset

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

After the reset exception handling is executed, the program branches off to the interrupt service routine. The address with which the interrupt service routine starts is stored in 0x0000_0004H

(Note 1) Set the $\overline{\text{RESET}}$ pin to "0" before turning the power on. Release the reset after the power supply voltage has stabilized sufficiently within the operating range.

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

4 Dubug Interface

4.1 Specification Overview

The TMPM380/M382 contains the Serial Wire JTAG Debug Port (SWJ-DP) unit for interfacing with the In-Circuit Emulator (ICE) and the Embedded Trace Macrocell™ (ETM) unit for instruction trace output. Trace data is output to the dedicated pins (TRACEDATA[0]-[1], SWV) via the on-chip Trace Port Interface Unit (TPIU).

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

4.2 Features of SWJ-DP

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

4.3 Features of ETM

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

4.4 Pin Functions

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

Table 4-1 SWJ-DP、ETM function

SWJ-DP Pin name	Name of port	JTAG debug function		SW debug	
		I/O	Discription	I/O	Discription
TMS/SWDIO	PB3	Input	JTAG Test Mode Selection	I/O	Serial Wire Data Input/Output
TCK/SWCLK	PB4	Input	JTAG Test Check	Input	Serial Wire Clock
TDO/SWV	PB5	Output	JTAG Test Data Output	— (Output) (Note)	(Serial Wire Viewer Output)
TDI	PB6	Input	JTAG Test Data Input	—	—
$\overline{\text{TRST}}$	PB7	Input	JTAG Test $\overline{\text{RESET}}$	—	—
TRACECLK	PB0	Output	TRACE Clock Output		
TRACEDATA0	PB1	Output	TRACE DATA Output0		
TRACEDATA1	PB2	Output	TRACE DATA Output1		

(Note) In case of enabling SWV function.

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

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

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

Initial Setting	PORT (Bit name)	Debug Function	Port Setting After Reset				
			Function (PBFR)	Input (PBIE)	Output (PBCR)	Pull-up (PBPUP)	Pull-down (PBPDN)
PORT	PB0	TRACECLK	0	0	0	0	0
PORT	PB1	TARCEDATA0	0	0	0	0	0
PORT	PB2	TRACEDATA1	0	0	0	0	0
DEBUG	PB3	TMS/SWDIO	1	1	1	1	0
DEBUG	PB4	TCK/SWCLK	1	1	0	0	1
DEBUG	PB5	TDO/SWV	1	0	1	0	0
DEBUG	PB6	TDI	1	1	0	1	0
DEBUG	PB7	$\overline{\text{TRST}}$	1	1	0	1	0

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

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

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

4.5 Connection with a Debug Tool

4.5.1 How to connect

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

4.5.2 When use general purpose port

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

Table 4-3 Debug Interface

	Using Debug Interface pins							
	$\overline{\text{TRST}}$	TDI	TDO/ SWV	TCK/ SWCLK	TMS/ SWDIO	TRACE DATA1	TARCE DATA0	TRACE CLK
JTAG+SW (After RESET)	○	○	○	○	○	×	×	×
JTAG+SW (No $\overline{\text{TRST}}$)	×	○	○	○	○	×	×	×
JTAG+TRACE	○	○	○	○	○	○	○	○
SW	×	×	×	○	○	×	×	×
SW+SWV	×	×	○	○	○	×	×	×
Disable Debug function	×	×	×	×	×	×	×	×

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

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

When Cortex-M3 CPU core going into HALT mode by break operation during debbug mode,Watch dog timer(WDT) count stops automatically. Other peripherals continue operation.

(Note) 16-bit timer (TMRB) and Multi purpose timer (MPTs) can be desable in HALT mode. Please refer to “16-bit timer (TMRB)” and “16-bit Multi-purpose Timers (MPTs)”

5 Memory Map

The memory maps for the TMPM380 are based on the ARM Cortex-M3 processor core memory map. The internal ROM, internal RAM and internal I/O of the TMPM380 are mapped to the code, SRAM and peripheral regions of the Cortex-M3 respectively. The SRAM and internal I/O regions are all included in the bit-band region.

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

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

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

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

5.1 TMPM380FY Memory Map

Fig 5-1 shows the memory map of the TMPM380FY.

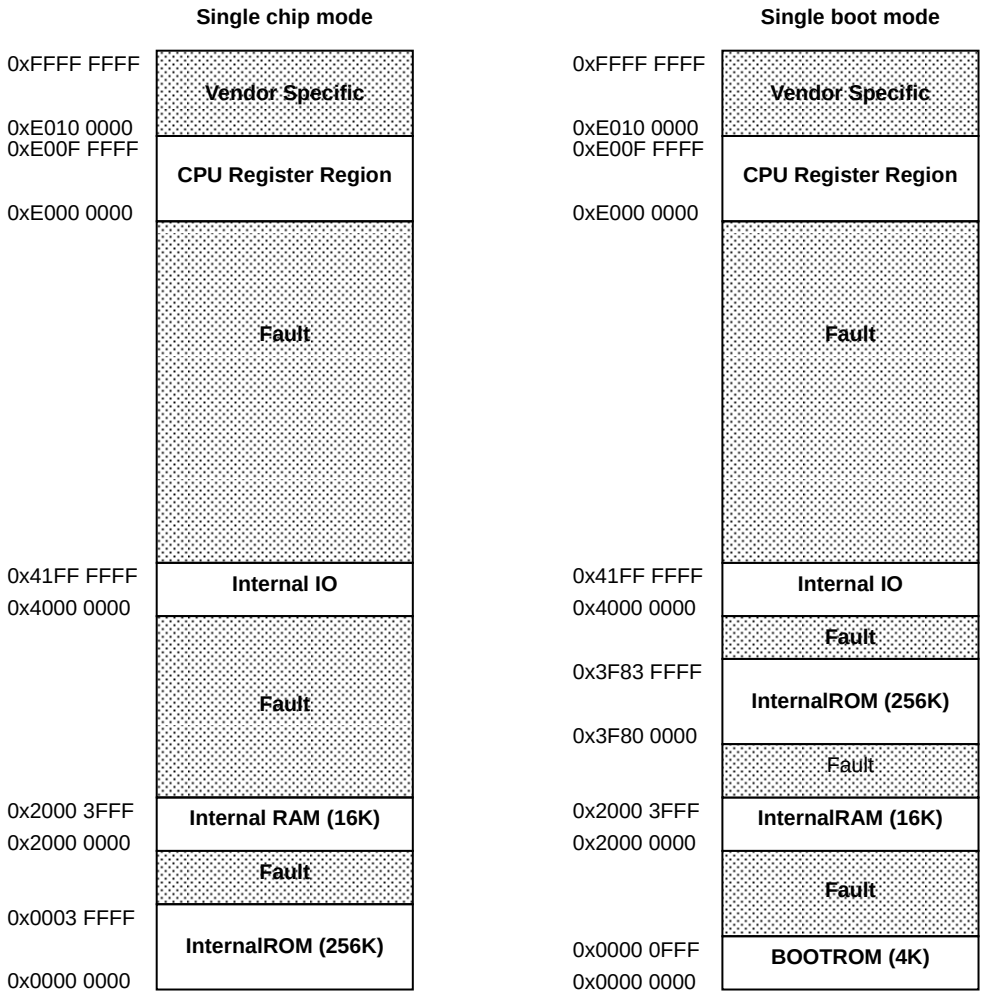


Fig 5-1 Memory Map

5.2 TMPM380FW/382FW Memory Map

Fig 5-2 shows the memory map of the TMPM380FW/382FW.

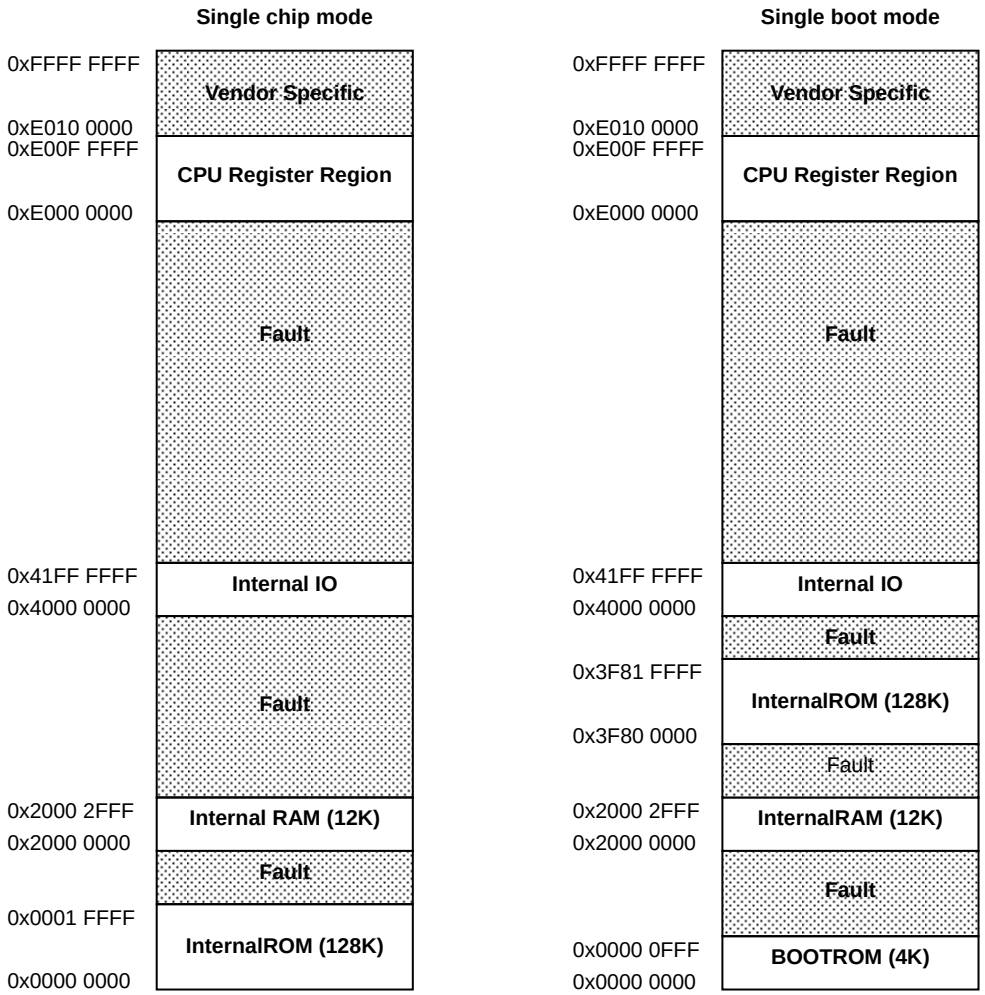


Fig 5-2 Memory Map

5.3 TMPM382FS Memory Map

Fig 5-3 shows the memory map of the TMPM382FS.

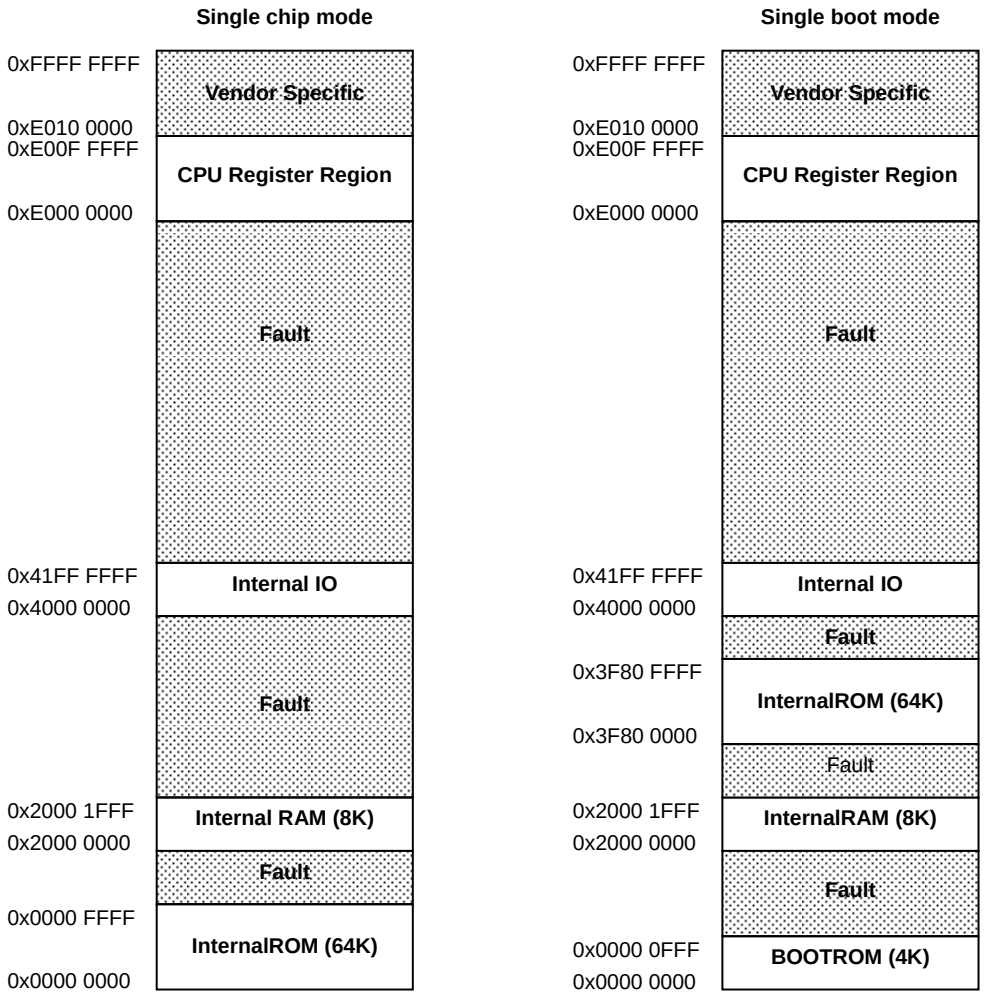


Fig 5-3 Memory Map

6 Clock/Mode Control

6.1 Features

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

The low power consumption mode can reduce power consumption.by mode transitions.

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

The clock/mode control block has the following functions:

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

In addition to NORMAL mode, the TMPM380/M382 can operate in three types of low power mode to reduce power consumption according to its usage conditions.

6.2 Registers

6.2.1 Register List

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

Table 6-1 Registers of Clock Generator

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

6.2.2 Detailed Description of Registers

6.2.2.1 System Control Register (CGSYSCR: 0x4004_0200)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	GEAR2	GEAR1	GEAR0
Read/Write	R					R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.					High-speed clock (fc) gear 000: fc 100: fc/2 001: reserved 101: fc/4 010: reserved 110: fc/8 011: reserved 111: fc/16		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	FPSEL1	FPSEL0	-	PRCK2	PRCK1	PRCK0
Read/Write	R		R/W		R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.		fperiph clock 00:fgear 01:fc 1*: fs		"0" is read.	Prescaler clock 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved		
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	FCSTOP	-	-	SCOSEL1	SCOSEL0
Read/Write	R/W	R		R/W	R		R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	Wtite "0"	"0" is read.		fclk for ADC 0:enable 1:disable	"0" is read.		SCOUT clock 00:fs 01:fsys/2 10:fsys 11:φT0	
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

<Bit 2:0><GEAR 2:0> : Specifies the high-speed clock (fc) gear.
<Bit 10:8><PRCK 2:0> : Specifies the prescaler clock to peripheral I/O.
<Bit 13:12><FPSEL 1:0> : Specifies the source clock to fperiph.
<Bit 17:16><SCOSEL 1:0> : Specifies the source clock to SCOUT.
<Bit 20><FCSTOP> : Specifies the fclk to ADC.
<Bit 23> : Must be write "0".

6.2.2.2 Oscillation Control Register (CGOSCCR: 0x4004_0204)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W
After reset	0	0	1	1	0	0	0	0
Function	Write “0011”				Clock source for Warm-up timer(WUP) 0: High-speed oscillator (follow to WUPSEL2) 1: Low-speed oscillator	PLL operation 0: Stop 1: Oscillation	Status of Warm-up timer (WUP) 0: warm-up completed 1: Warm-up in operation	Operation of warm-up timer (WUP) 0: don’t care 1: starting warm-up
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	WUODR1	WUODR0	-	-	-	-	XTEN	XEN1
Read/Write	R/W		R/W	R/W	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Bit1:0 for warm-up counter value. If high-speed oscillator is selected,<Bit15:14><WOUDR 1:0> is set “0”.		Write “0”		“0” is read.		Low-speed oscillator (External) 0: Stop 1: Oscillation	High-speed oscillator 1 (External) 0: Stop 1: Oscillation
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	WUODR5	WUODR4	WUODR3	WUODR2	WUPSEL2	HOSCON	OSCSEL	XEN2
Read/Write	R/W				R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	Bit5:2 for warm-up counter value.				Clock source for Warm-up timer(WUP) 0: Internal (OSC2) 1: External (OSC1)	Port M or X1/X2 0: PORT M 1: X1/X2	Selection of high-speed oscillator 0: Internal (OSC2) 1: External (OSC1)	High-speed oscillator 2 (Internal) 0: Stop 1: Oscillation
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	WUODR13	WUODR12	WUODR11	WUODR10	WUODR9	WUODR8	WUODR7	WUODR6
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Bit13:6 for warm-up counter value.							

<Bit 0><WUEON> : Enables to start the warm-up timer.
“0” is read.

<Bit 1><WUEF> : Enables to monitor the status of the warm-up timer.

<Bit 2><PLLON> : Specifies operation of the PLL.
It stops after reset. Setting the bit is required.

Note : When using the internal oscillator,please do not use PLL.

<Bit 3><WUPSEL1> : Select source clock for warm-up timer between high-speed oscillator and low-speed oscillator. High-speed oscillator is followed by <Bit19><WUPSEL2>.

<Bit 8><XEN1> : Specifies operation of the external high-speed oscillator1.

<Bit 9><XTEN> : Specifies operation of the external low-speed oscillator.

<Bit 16><XEN2> : Specifies operation of the internal high-speed oscillator2.

<Bit 17><OSCSEL> : High speed oscillator switch from internal oscillator(OSC1) to external oscillator(OSC2).OFD does not use the feature, after switching the external oscillator, it is recommended to stop the internal oscillator to reduce power consumption. Also, after switching to an external oscillator, please not switch to the internal oscillator.

<Bit 18><HOSCON> : Specifies Port M or X1/X2. When external oscillator is used, set PMCR/PMPUP /PMPDN/PMIE of Port M to disable. After reset, PMCR/PMPUP/PMPDN/PMIE are set to disable.

<Bit 19><WUPSEL2> : Select source clock for warm-up timer between internal oscillator(OSC1) and external oscillator(OSC2). When using STOP/SLEEP mode, please select clock-surce that is same as <OSCSEL> to <WUPSEL2> before entering to STOP/SLEEP mode.

<Bit 31:24, 23:20, 15:14><WUODR 13:0> : Warm-up timer value.

※If high-speed oscillator is selected, <Bit 15:14><WUODR 1:0> is set “00”.

<Bit 7:4> : Must be write “0011”.

<Bit 13:12> : Must be write “00”.

6.2.2.3 Standby Control Register (CGSTBYCR: 0x4004_0208)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	STBY2	STBY1	STBY0
Read/Write	R					R/W	R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function	"0" is read.					Low power consumption mode 000: Reserved 001: STOP 010: SLEEP (Note) 011: IDLE 1**: Reserved		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	RXTEN	RXEN
Read/Write	R						R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	"0" is read.						Low-speed oscillator after releasing STOP mode 0: Stop 1: Oscillation	High-speed oscillator after releasing STOP mode 0: Stop 1: Oscillation
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	DRVE
Read/Write	R				R/W			R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.				Write "000".			Port drive in STOP mode 0: Hi-Z 1: Drive
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

<Bit 2:0><STBY2:0> : Specifies the low power consumption mode.

Note: Before switch to SLOW mode, if the low-speed oscillator is disabled, enable the low-speed oscillator and then activate the warm-up by software. And msut be selected "fperiph" for Peripferal I/O clock as fs by CGCYSCR<FPSEL[1:0]>=1*.

<Bit 8><RXEN> : Specifies the high-speed oscillator operation after releasing the STOP mode. Must be written same data of CGOSCCR<XEN1> or <XEN2> which is selected to fc.

<Bit 9><RXTEN> : Specifies the Low-speed oscillator operation after releasing the STOP mode. Must be written same data of CGOSCCR<XTEN>.

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

<Bit 19:17> : Must be written "000".

6.2.2.4 PLL Selection Register (CGPLLSEL: 0x4004_020C)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	-	PLLSEL
Read/Write	R/W							R/W
After reset	0	0	0	1	1	1	1	0
Function	Write “0001111”							Select PLL output 0: fosc 1: fpll
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R/W				R	R/W		
After reset	0	1	1	1	0	0	1	0
Function	Write “0111”				“0” is read.	Write “010”		
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read							
	31	**30**	**29**	**27**	**26**	**25**	**24**	**23**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read							

<Bit 0><PLLSEL> : Specifies use or disuse of the clock multiplied by the PLL.
“fosc” is default value after reset. Setting is required when using the PLL.
<Bit 7:1> : Must be write “0001111”
<Bit 10:8> : Must be write “010”
<Bit 15:12> : Must be write “0111”

6.2.2.5 System Clock Selection Register (CGCKSEL: 0x4004_0210)

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

<Bit 0>< SYSCKFLG > : Shows the status of the system clock. Switching the reading oscillator with <SYSCK> generates time lag to complete. If the read value from <SYSCKFLG> is the same as the value specified in <SYSCK>, the switching has been completed.
<Bit 1><SYSCK> : Specifies system clock. When change value of this bit, oscillation must stable High-speed oscillator(1 or 2) and Low-speed oscillator. According to the used oscillator, corresponding CGOSCCR<XEN1>, <XEN2> or <XTEN> must be set to "1" in advance.

6.3 Clock Control

6.3.1 Clock System Block Diagram

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

fosc1 : Clock input from external high-speed oscillator (X1 and X2)
fosc2 : Clock input from internal high-speed oscillator
fs : Clock input from external low-speed oscillator (XT1 and XT2)
fosc : High-speed clock specified by CGOSCCR<OSCSEL>
fpll : Clock quadrupled by PLL
fc : High-speed clock specified by CGPLLSEL<PLLSEL>
fgear : High-speed clock specified by CGSYSCR<GEAR2:0>
fsys : The same clock as fgear (system clock)
fperiph : Clock specified by CGSYSCR<FPSEL1:0>
ΦT0 : Prescaler clock specified by CGSYSCR<PRCK2:0>

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

- 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 initializes the clock configuration as follows.

High-speed oscillator2 (Internal) : ON (oscillating)
High-speed oscillator1 (External) : OFF (stop) X1,X2
Low-speed oscillator (External) : OFF (stop) XT1,XT2
PLL (phase locked loop circuit) : OFF (stop)
High-speed clock gear : fc (no frequency dividing)

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

fc = fosc2
fsys = fc (=fosc2)
fperiph = fc (=fosc2)
ΦT0 = fperiph (=fosc2)

Reset configures fsys to high-speed oscillator2 (internal)..

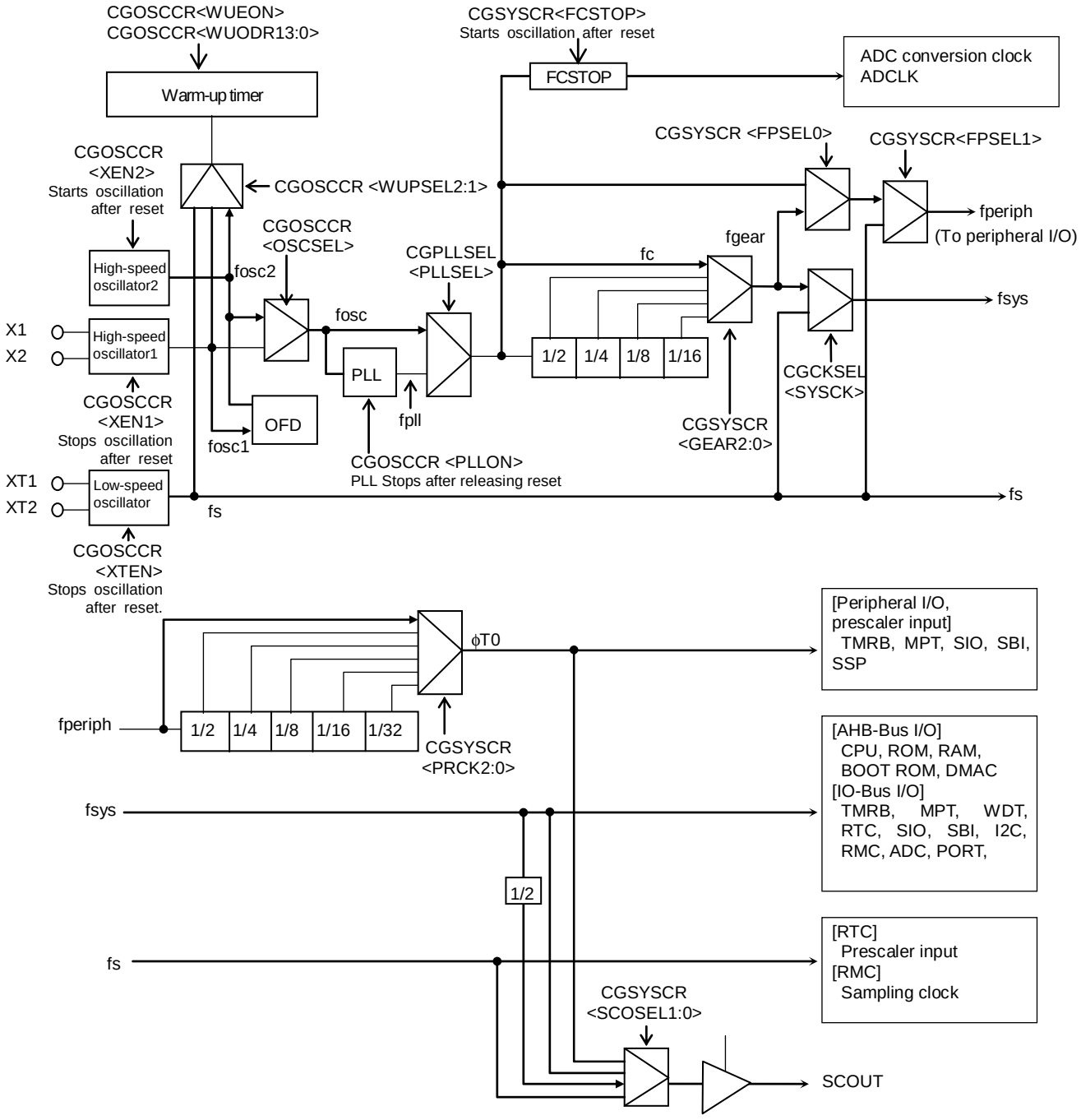


Fig. 6-1 Clock Block Diagram

(Note1)	**The input clocks to selector shown with an arrow are set as default after reset.**
(Note2)	**OFD can detect oscillation frequency of fosc1 only.**

6.3.3 Clock Multiplication Circuit (PLL)

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

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

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

6.3.4 Warm-up Function

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

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

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

How to configure the warm-up function

Specify the count up clock for the warm-up timer in the CGOSCCR<WUPSEL 2:1> bits.

The warm-up time can be specified by setting the CGOSCCR<WUODR13:0>. The CGOSCCR<WUEON><WUEF> are used to confirm the start and completion of warm-up timer through software (instruction). After the completion of warming-up timer is confirmed, switch the system clock by setting the CGCKSEL<SYSCK>.

When clock switching occurs, the current system clock can be checked by monitoring the CGCKSEL<SYSCKFLG>.

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

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

<Example Securing the stability time for the PLL(In case of high-speed oscillator1) >

CGOSCCR<WUPSEL1>=”0” : Specify High speed oscillator.
CGOSCCR<WUPSEL2>=”1” : Specify External oscillator.
CGOSCCR<WUPODR13:0>=” warm-up time (1/fosc1)/4”/
: Specify the warm-up time
CGOSCCR0<WUEON>=”1” : Start the warm-up timer (WUP)
CGOSCCR0<WUEF> Read : Wait until the state becomes "0" (warm-up is finished)

(Note) When high-speed oscillator is selected by setting (<WUPSEL2:1>=”＊0 “) for count-up clock of warming-up timer, lower 2-bits of <WUODR13:0> is “00”.

6.3.5 System Clock

The TMPM380 offers three selectable system clocks : two high-speed clocks and one low-speed clock.

Two kinds of high-speed clocks are selectable either internal oscillator or external oscillator. After reset, internal oscillator is available, and external oscillator is stop. The high-speed clocks are dividable.

6.3.5.1 High speed clock

- Input frequency from high-speed oscillator1 (X1 and X2) : 8MHz to 10MHz
- Input frequency from high-speed oscillator2 (internal oscillator) : 10MHz
- Clock gear:1/1, 1/2, 1/4, 1/8, 1/16 (after reset: 1/1)

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

Input frequency		Min. operating freq.	Max. operating freq.	After reset (PLL=OFF, CG=1/1)	Clock gear (PLL=ON)					Clock gear (PLL=OFF)				
					1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
osc1	8MHz	1MHz	40MHz	8MHz	32	16	8	4	2	8	4	2	1	**
	10MHz			10MHz	40	20	10	5	2.5	10	5	2.5	1.25	**
osc2	10MHz			10MHz	40	20	10	5	2.5	10	5	2.5	1.25	**

(Note1) PLL=ON/OFF setting: available in CGOSCCR<PLLON>

(Note2) Clock gear setting: available in CGSYSCR<GEAR2:0>
Switching of clock gear is executed when a value is written to the CGSYSCR<GEAR2:0> register. The actual switching takes place after a slight delay.

(Note3) Do not select 1/16 of clock gear when PLL is set off.

(Note4) ** :Reserved

The following are the procedure of switching over from the internal oscillator(OSC2) to the external oscillator(OSC1).

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

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

With clearing CGOSCCR<HOSCON> to “0”, rewriting the CGOSCCR<XEN1> is prohibited.

6.3.5.2 Low speed clock

- Input frequency from XT1 and XT2

Table 6-3 Range of Low Frequency

Input Frequency Range	Maximum Operating Frequency	Minimum Operating Frequency
30 to 34(kHz)	34 kHz	30 kHz

6.3.6 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<FPSEL1:0> can be divided according to the setting in the CGSYSCR<PRCK2:0>. After the controller is reset, fperiph 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.

6.3.7 System Clock Pin Output Function

The TMPM380 enables to output the system clock from a pin. The PA1/SCOUT pin can output the low speed clock fs, the system clock fsys and fsys/2, and the prescaler input clock for peripheral I/O φT0. By setting the port A registers, the PACR<PA1C> and PAFR2<PA1F2> to “1”, the PA1/SCOUT pin becomes the SCOUT output pin. The output clock is selected by setting the CGSYSCR<SCOSEL1:0>.

Table 6-4 shows the pin states in each mode when the SCOUT pin is set to the SCOUT output.

Table 6-4 Scout Output State in Each Mode

Mode / SCOUT selection CGSYSCR	NORMAL	SLOW	Low power consumption mode		
			IDLE	SLEEP	STOP
<SCOSEL1:0> = “00”	Output the fs clock.				
<SCOSEL1:0> = “01”	Output the fsys/2 clock.				
<SCOSEL1:0> = “10”	Output the fsys clock.				
<SCOSEL1:0> = “11”	Output the ΦT0 clock.		Fixed to “0” or “1”.		

(Note) The phase difference (AC timing) between the system clock output by the SCOUT and the internal clock is not guaranteed.

6.4 Modes and Mode Transitions

6.4.1 Mode Transitions

The NORMAL mode and the SLOW mode use the high-speed and low-speed clocks for system clock respectively.

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

When the low-speed clock is not used, the SLOW and SLEEP modes cannot be used.

Fig. 6-2 shows a mode transition diagram

For a description of sleep-on-exit, refer to “Cortex-M3 Technical Reference Manual”.

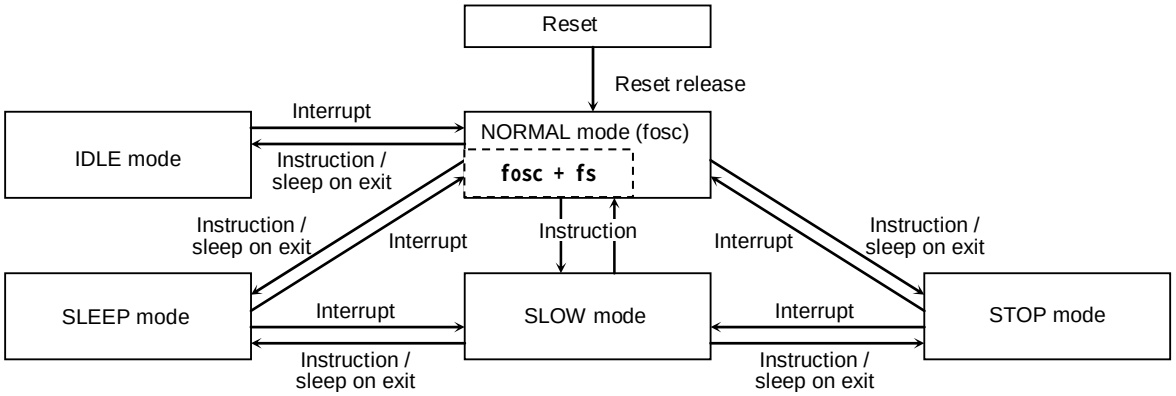


Fig. 6-2 Mode Transition Diagram

6.5 Operation Modes

Two operation modes, NORMAL and SLOW, are available. The features of each mode are described below.

6.5.1 NORMAL Mode

This mode is to operate the CPU core and the peripheral hardware by using the high-speed clock(fosc1 or fosc2). It is shifted to the NORMAL mode with fosc2(internal high-speed oscillator) after reset. Also low-speed clock is possible to use.

6.5.2 SLOW Mode

This mode is to operate the CPU core and the peripheral hardware by using the low-speed clock with high-speed clock stopped.
The SLOW mode reduces power consumption compared to the NORMAL mode.
This mode allows only the following peripheral functions to operate: I/O ports, real-time clock (RTC), TMRB, MPT(TMRB mode), remote control signal preprocessor (RMC).

(Note1)	**Be sure to stop peripheral functions except for the CPU, RTC, I/O ports, TMRB, MPT(TMRB mode) and RMC before switching to the SLOW mode.**
(Note2)	**In the slow mode, be sure not to perform reset using the Application Interrupt and Reset Control Register<SYSRESETREG> of the Cortex-M3 NVIC register.**

6.5.3 Low Power Consumption Mode

The TMPM380 has three low power consumption modes: IDLE, SLEEP and STOP. To switch to the low power consumption mode, specify the mode in the system control register CGSTBYCR<STBY2:0> and execute the WFI (Wait For Interrupt) instruction. In this case, execute reset or generate the interrupt to release the mode. Releasing by the interrupt requires settings in advance.

(Note 1)	**Transition to the low power consumption mode by executing the WFE (Wait For Event) instruction is prohibited as the TMPM380 does not offer any event for releasing the low power consumption mode.**
(Note 2)	**The TMPM380 does not support the low power consumption mode configured with the SLEEPDEEP bit in the Cortex-M3 core. Setting the SLEEPDEEP bit of the system control register is prohibited.**

The features of each mode are described as follows.

6.5.4 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)
- 16-bit multi purpose timer counter (MPT : except PMD function)
- Serial channel (SIO)
- Serial bus interface (SBI)
- AD converter (ADC)
- Watchdog timer (WDT)
- PLL

6.5.5 SLEEP Mode

The internal low-speed oscillator, real time clock and RMC can operate. By releasing the SLEEP mode, the device returns to the preceding mode of the SLEEP mode and starts operation.

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

6.5.6 STOP Mode

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

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

The STOP mode enables to select the pin status by setting the CGSTBYCR<DRVE>.Table 6-5 shows the pin status in the STOP mode.

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

Table 6-5 Pin States in STOP Mode

	Pin Name	I/O	<DRVE>=0	<DRVE>=1
control pins	$\overline{\text{RESET}}$, MODE	Input only	○	○
Oscillator	X1,XT1	Input only	×	×
	X2, XT2	Output only	“H” level output	“H” level output
Ports	PAx to PPx	Input	×	Depends on PxIE<n>.
		Output	×	Depends on PxCR<n>.
Debag Interface	TMS/SWDIO/TDO/SWV	Input	×	Depends on PxIE<n>
		Output	●	●
External Interrupts	INT0 to INTF	Input	Depends on PxIE<n>.	Depends on PxIE<n>.
SSP	SPnCLK/SPnFSS/SPnDO	Output	×	●
MPT(IGBT)	$\overline{\text{GEMGn}}$/MTnIN	Input	×	Depends on PxIE<n>.
	MTnOUTxx	Output	●	●
MPT(PMD)	$\overline{\text{EMGn}}$	Input	×	Depends on PxIE<n>.
	UOn/VOn/WOn/XOn/YOn/ZOn	Output	●	●
Others		Input	×	Depends on PxIE<n>.
		Output	×	Depends on PxCR<n>.

○ : Input or output enabled

× : Input or output disabled.

● : **OE signal (internal) of selected function is enabled and PxCR<n>=1, then signal is output.**

n : Bit number

6.5.7 Low power Consumption Mode Setting

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

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

Table 6-6 Low power consumption mode setting

Mode	CGSTBYCR <STBY2:0>
STOP	001
SLEEP	010
IDLE	011

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

6.5.8 Operational State in Each Mode

Table 6-7 show the operational state in each mode.

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

Table 6-7 Operational State in Each Mode

Block	NORMAL	SLOW	IDLE	SLEEP	STOP
Processor core	○	○	×	×	×
I/O port	○	○	○	○	* (Note 3)
SSP	○	× (Note 1)	×	×	×
ADC	○	× (Note 1)	ON/OFF selectable for each module	×	×
SIO	○	× (Note 1)		× (Note 1)	× (Note 1)
SBI	○	× (Note 1)		× (Note 1)	× (Note 1)
WDT	○	× (Note 1)		× (Note 1)	× (Note 1)
TMRB	○	○		× (Note 1)	× (Note 1)
MPT(TMRB mode)	○	○		× (Note 1)	× (Note 1)
MPT(IGBT mode)	○	× (Note 1)		× (Note 1)	× (Note 1)
MPT(PMD mode)	○	× (Note 1)	○	× (Note 1)	× (Note 1)
RMC	○	○	○	○	×
RTC	○	○	○	○	×
CG	○	○	○	○	×
PLL	○	×	○	×	×
OFD	○(Note 4)	×	○(Note 4)	×	×
High-speed oscillator 1 (fosc1)	○	×(Note 2)	○	×	×
High-speed oscillator 1 (fosc2)	○	×(Note 2)	○	×	×
Low-speed oscillator (fs)	○	○	○	○	×

○: Operating, ×: Stopped

(Note 1) In the SLOW mode, the ADC, SIO, SBI, SSP, MPT(IGBT,PMD) and WDT cannot be used and must be stopped before switch to SLOW mode.

(Note 2) The high-speed oscillator(1 or 2) does not stop automatically in SLOW mode and must be stopped by setting CGOSCCR1<XEN1> or <XEN2> after switched to SLOW mode.

The high-speed oscillator(1 or 2) does not oscillate automatically in SLOW mode and must be enabled by setting CGOSCCR<XEN1> or <XEN2> before switch to NORMAL mode.

(Note 3) The state depends on the CGSTBYCR<DRVE>.

(Note 4) When selecting fosc2 to system clock, OFD can't be use.

6.5.9 Releasing the Low Power Consumption Mode

The low power consumption mode can be released by an interrupt request, NMI or reset. The release source that can be used is determined by the low power consumption mode selected. Details are shown in Table 6-8.

Table 6-8 Release Source in Each Mode

Low power consumption mode			IDLE	SLEEP	STOP
Release source	Interrupt	INT0~F (Note 1)	○	○	○
		INTRTC	○	○	×
		INTRMCRX	○	○	×
		INTSSP0,1	×	×	×
		INTSBI0,1	○	×	×
		INTRX0 to 4,/ INTTX0 to 4	○	×	×
		INTADPD0,1/ INTADCP0,1	○	×	×
		INTADTMR/ INTADSFT	○	×	×
		INTPMD0,1/ INTEMG0,1	○	×	×
		INTMTTB00 to 20	○	×	×
		INTMTTB01 to 21	○	×	×
		INTMTCAP00 to 20, 01,to 21	○	×	×
		INTMTEMG0,1,2	○	×	×
		INTTB00 to 70,01 to 71	○	×	×
		INTCAP00 to 70, 01 to 71	○	×	×
		INTENC0,1	○	×	×
		INTDMACERR/ INTDMACTC	○	×	×
	NMI (INTWDT)		○	×	×
	NMI (INTVLTD)		○	○	○
	RESET (RESET pin and POR)		○	○	○

○: 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 switching to the low power consumption mode, set the CPU to prohibit all the interrupts other than the release source. If not, releasing may be executed by an unspecified interrupt.**

- Release by interrupt request

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

- Release by NMI

There are two kinds of NMI sources: WDT interrupt (INTWDT) and VLTD interrupt (INTVLTD). INTWDT can be used in IDLE mode only.

● Release by reset

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

Refer to section of “Interrupts" for details.

6.5.10 Warm-up

Mode transition requires the warm-up so that the oscillator provides stable oscillation.

In the mode transition from STOP to NORMAL/ SLOW or from SLEEP to NORMAL, the warm-up counter is activated automatically. And then the system clock output is started after the elapse of configured warm-up time. It is necessary to select the oscillator to be used for warm-up in the CGOSCCR<WUPSEL2:1> and to set the warm-up time in the CGOSCCR<WUODR13:0> before executing the instruction to enter the STOP/ SLEEP mode.

(Note) In STOP/ SLEEP modes, the PLL is disabled. When returning from these modes, configure the warm-up time in consideration of the stability time of the PLL and the oscillator. It takes approx. 200μs for the PLL to be stabilized.

In the transition from NORMAL to SLOW/ SLEEP, the warm-up is required so that the internal oscillator to stabilize if the low-speed oscillator is disabled. Enable the low-speed oscillator and then activate the warm-up by software.

In the transition from SLOW to NORMAL when the high-speed oscillator is disabled, enable the high-speed oscillator and then activate the warm-up.

Shows whether the warm-up setting of each mode transition is required or not.

Table 6-9 Warm-up setting in mode transition

Mode transition	Warm-up setting
NORMAL→IDLE	Not required
NORMAL→SLEEP	(Note 1)
NORMAL→SLOW	(Note 1)
NORMAL→STOP	Not required
IDLE→NORMAL	Not required
SLEEP→NORMAL	Auto-warm-up
SLEEP→SLOW	Not required
SLOW→NORMAL	(Note 2)
SLOW→SLEEP	Not required
SLOW→STOP	Not required
STOP→NORMAL	Auto-warm-up
STOP→SLOW	Auto-warm-up

(Note 1) If the low-speed oscillator is disabled, enable the low-speed oscillator and then activate the warm-up by software.
(Note 2) If the high-speed oscillator is disabled, enable the high-speed oscillator and then activate the warm-up by software.

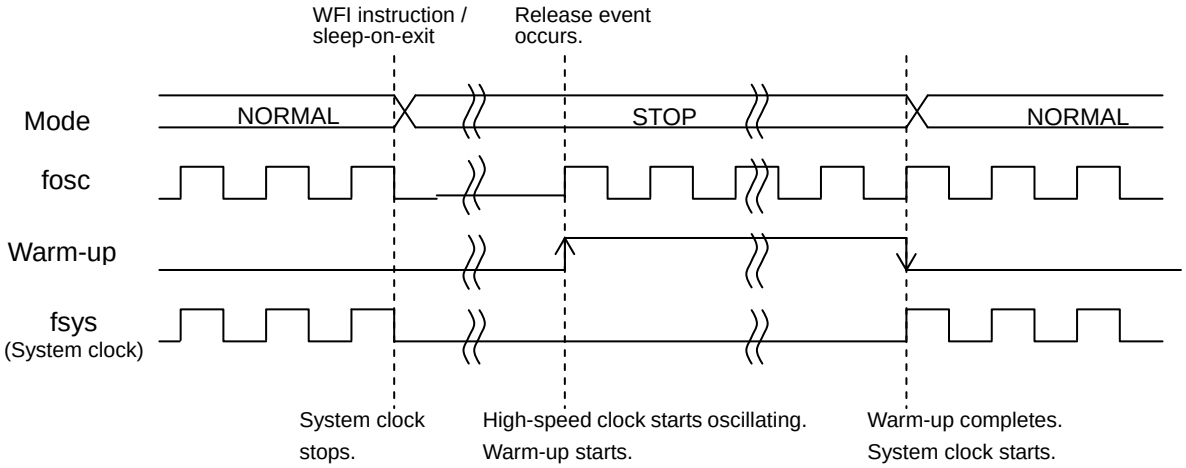
6.5.11 Clock Operations in Mode Transition

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

6.5.11.1 Transition of operation modes: NORMAL→STOP→NORMAL

Before entering to STOP mode, please set warming-up time to CGOSCCR<WUODR[13:0]> and select clock-source that is same as <OSCSEL> to <WUPSEL2>.

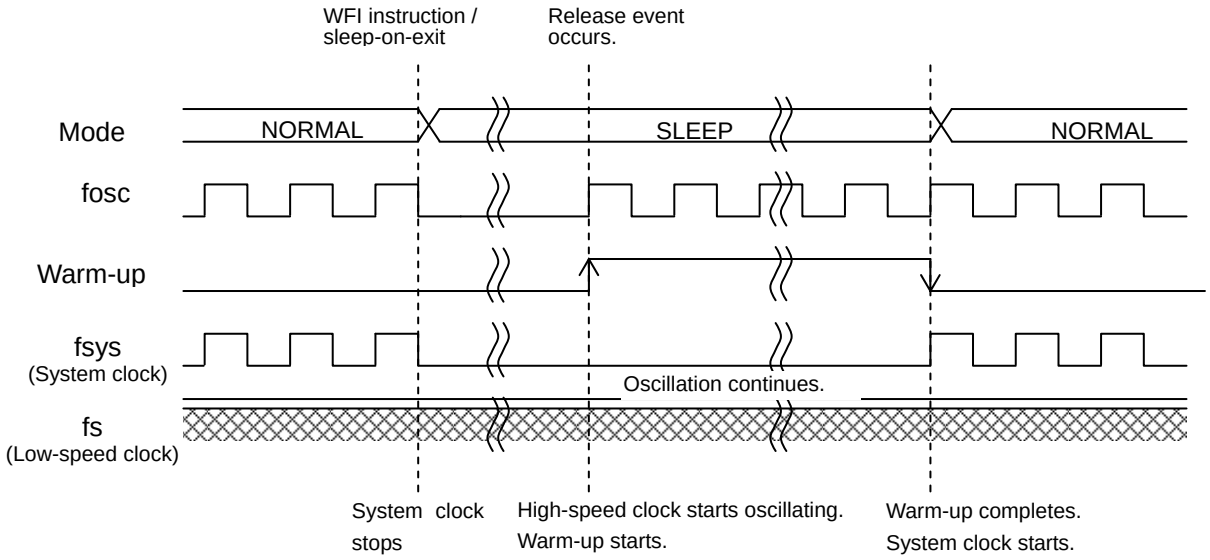
When returning to NORMAL mode from STOP mode, warming-up timer is started automatically.



6.5.11.2 Transition of operation modes: NORMAL→SLEEP→NORMAL

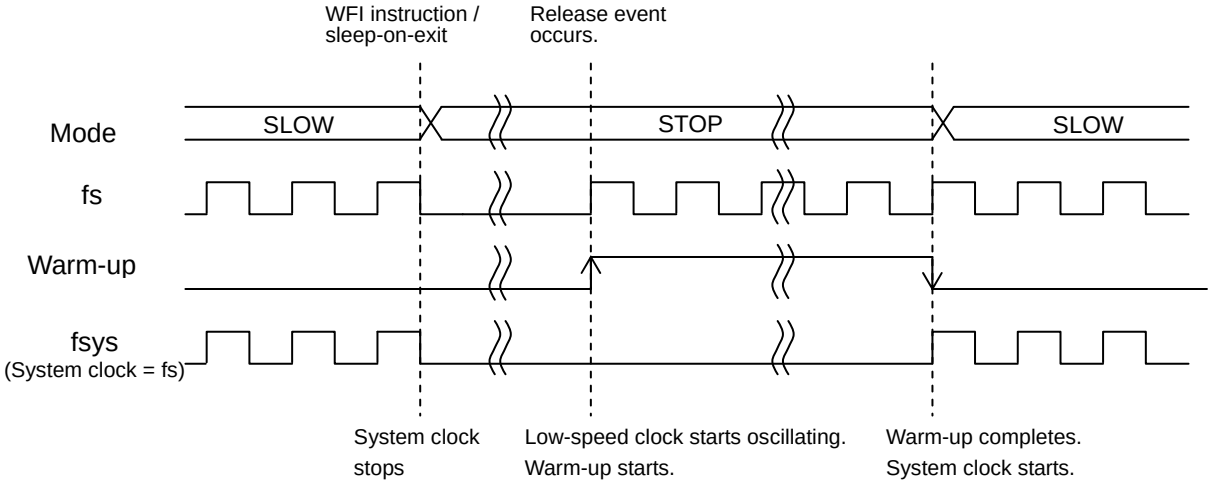
Before entering to SLEEP mode, please set warming-up time to CGOSCCR<WUODR[13:0]> and select clock-source that is same as <OSCSEL> to <WUPSEL2>.

When returning to NORMAL mode from SLEEP mode, warming-up timer is started automatically.



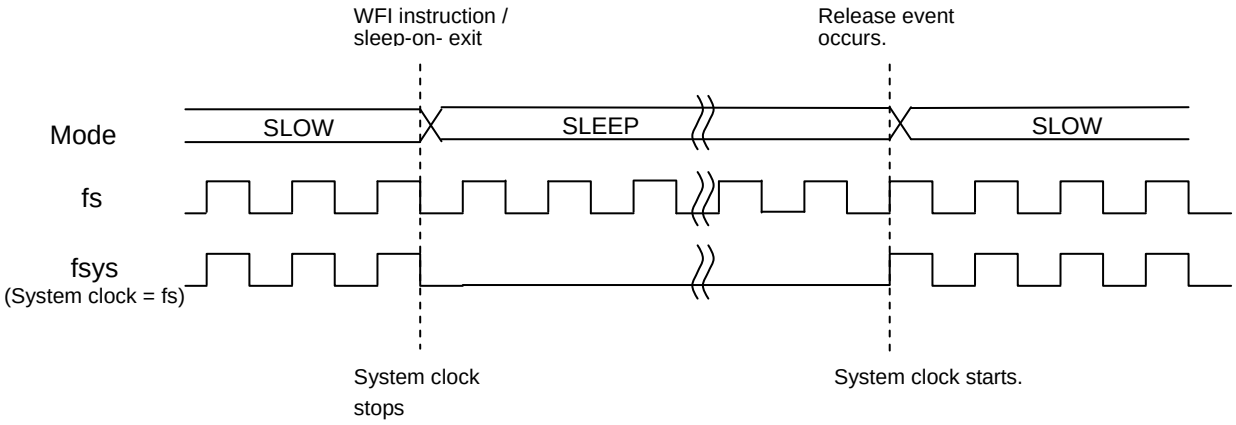
6.5.11.3 Transition of operation modes: SLOW→STOP→SLOW

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



6.5.11.4 Transition of operation modes: SLOW→SLEEP→SLOW

The low-speed clock continues oscillation in the SLEEP mode. There is no need to make a warm-up setting.



7 Exceptions

This chapter describes features, types and handling of exceptions.
Exceptions have close relation to the CPU core. Refer to “Cortex-M3 Technical Reference Manual” if needed.

7.1 Overview

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

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

7.1.1 Exception Types

The following types of exceptions exist in the Cortex-M3.
For detailed descriptions on each exception, refer to “Cortex-M3 Technical Reference Manual”.

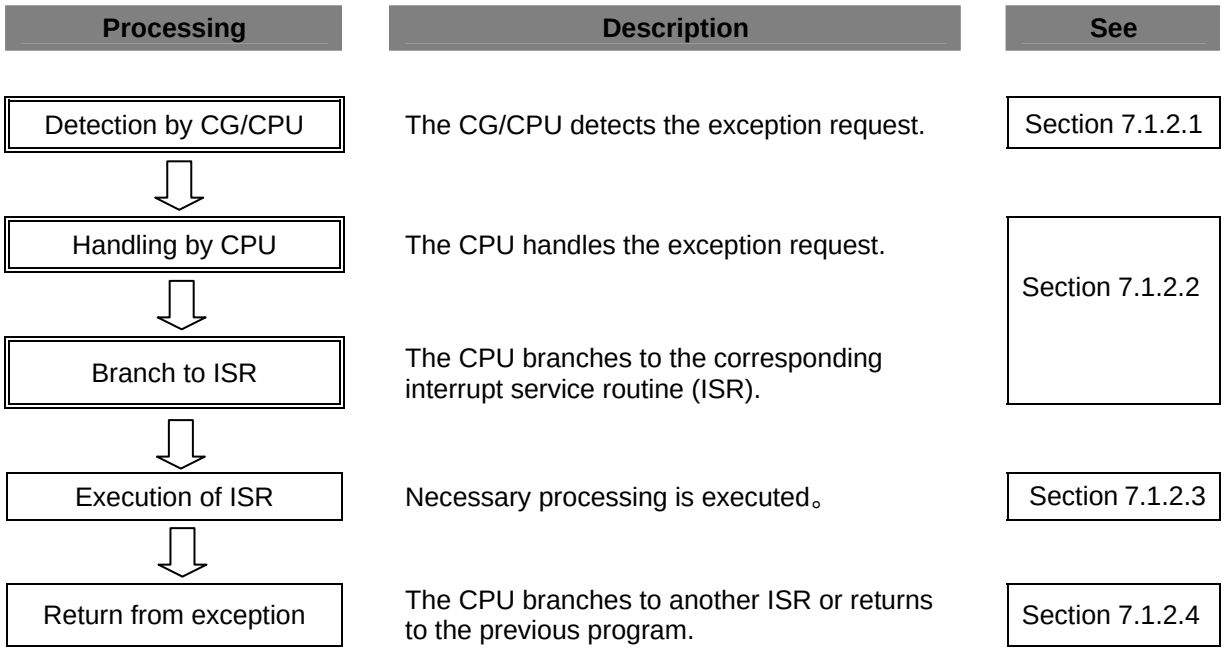
- Reset
- Non-Maskable Interrupt (NMI)
- Hard Fault
- Memory Management
- Bus Fault
- Usage Fault
- SVCall (Supervisor Call)
- Debug Monitor
- PendSV
- SysTick
- External Interrupt

7.1.2 Handling Flowchart

The following shows how an exception/interrupt is handled.

indicates hardware handling. Indicates software handling.

Each step is described later in this chapter.



7.1.2.1 Exception Request and Detection

(1) Exception occurrence

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

An exception by CPU instruction execution is caused when the CPU executes an instruction that generate an exception or when an error condition occurs during instruction execution.

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

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

(2) Exception detection

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

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

Table 7-1 Exception Types and Priority

No	Exception type	Priority	Description
1	Reset	-3 (highest)	Reset pin, WDT or SYSRETREQ
2	Non-Maskable Interrupt	-2	WDT or VLTD
3	Hard Fault	-1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4	Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) (Note 1) Instruction fetch from the Execute Never (XN) region
5	Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map
6	Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution
7-10	Reserved		
11	SVCall	Configurable	System service call with SVC instruction
12	Debug Monitor	Configurable	Debug monitor when the core is not halting
13	Reserved		
14	PendSV	Configurable	Pendable system service request
15	SysTick	Configurable	Notification from system timer
16-	External Interrupt	Configurable	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 "7.5.1.4 List of Interrupt Factors".**

(3) Priority setting

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

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

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

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

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

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

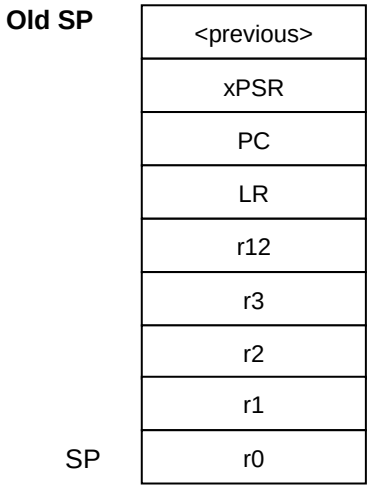
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called “pre-emption”.

(1) Stacking

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

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

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



(2) Fetching an ISR

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

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

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

(3) Late-arriving

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

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

(4) Vector table

The vector table is configured as shown below.

It must always be set the first four words (stack top address, reset ISR address, NMI ISR address, and Hard Fault ISR address) .
ISR addresses for other exceptions are prepared in case if necessary.

Offset	Exception	Contents	Setting
0x00	Reset	Initial value of the main stack	Required
0x04	Reset	ISR address	Required
0x08	Non-Maskable Interrupt	ISR address	Required
0x0C	Hard Fault	ISR address	Required
0x10	Memory Management	ISR address	Optional
0x14	Bus Fault	ISR address	Optional
0x18	Usage Fault	ISR address	Optional
0x1C ～ 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

7.1.2.3 Executing an ISR

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

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

For details about interrupt handling, see "7.5 Interrupt".

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

7.1.2.4 Exception exit

(1) Execution after returning from an ISR

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

- Tail-chaining

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

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

- Returning to the last stacked ISR

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

- Returning to the previous program

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

(2) Exception exit sequence

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

- Pop eight registers

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

- Load current active interrupt number

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

- Select SP

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

7.2 Reset Exceptions

Reset exceptions are generated from the following three sources.
Use the Reset Flag (CGRSTFLG) Register of the Clock Generator to identify the source of a reset.

・External reset pin
A reset exception occurs when an external reset pin changes from “L” to “H”.

・Reset exception by POR
A reset exception occurs when the power is turned on. For details, see the chapter on the POR.

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

・Reset exception by OFD
The oscillation frequency detection (OFD) has a reset generating feature. For details, see the chapter on the OFD.

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

7.3 Non-Maskable Interrupts (NMIs)

Non-maskable interrupts are generated from the following two sources.
Use the NMI Flag (CGNMIFLG) Register of the clock generator to identify the source of a non-maskable interrupt.

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

・Non-maskable interrupt by VLTD
The Voltage Level Detector (VLTD) has a non-maskable interrupt generating feature. For details, see the chapter on the VLTD.

7.4 SysTick

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

In case set a value in the SysTick Reload Value Register and enable the SysTick features in the SysTick Control and Status Register, the counter loads with the value set in the Reload Value Register and begins counting down. When the counter reaches "0", a SysTick exception occurs. It is also possible to pend exceptions and use a flag to know when the timer reaches "0".

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

(Note)	**In this product, the system timer counts based on a clock obtained by dividing the clock input from the X1 pin by 32.** **The SysTick Calibration Value Register is set to 0x9C4, which provides 10 ms timing when the clock input from X1 is 8 MHz. In case of 10MHz clock input, 10 ms timing is made by setting 0xC35 to SysTick Reload Register.**

7.5 Interrupt

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

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

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

7.5.1 Interrupt factors

7.5.1.1 Interrupt route

Fig 7-1 shows an interrupt request route.

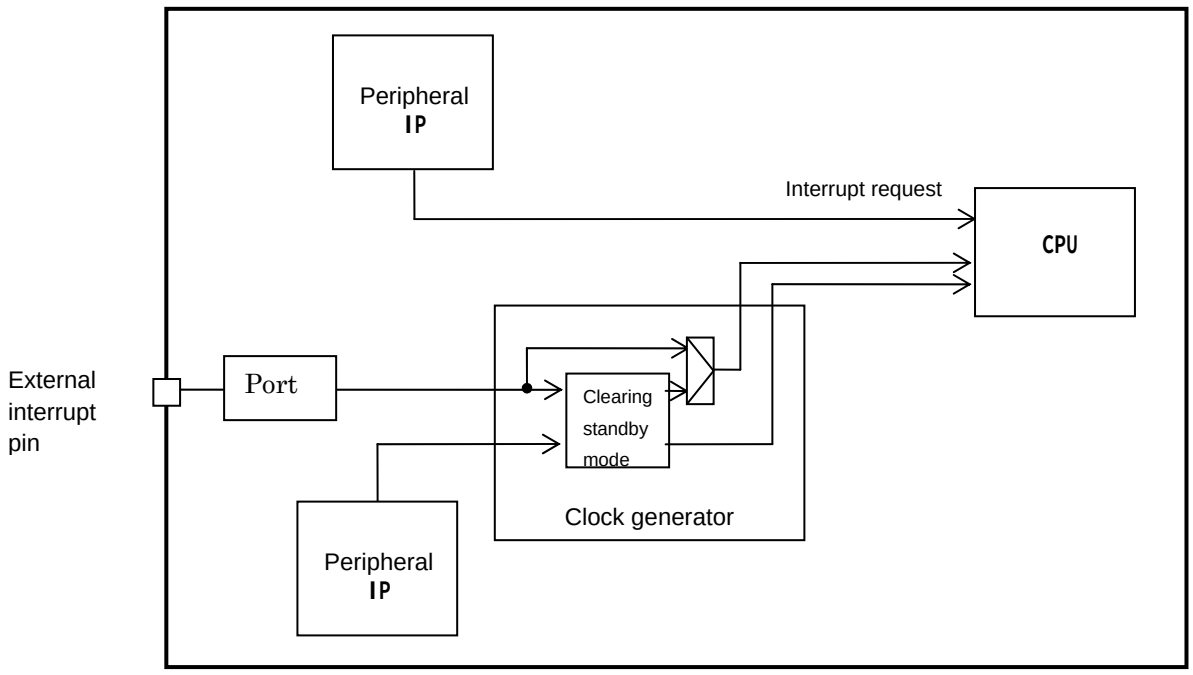


Fig 7-1 Interrupt Route

7.5.1.2 Interrupt signal generation

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

・From external pin

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

・From peripheral IP

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

See chapters of relevant IP for details.

7.5.1.3 Transmission of interrupt signal

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

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

7.5.1.4 List of Interrupt Factors

Table 7-2 shows the list of interrupt factors.

Table 7-2 List of Hardware Interrupt Factors (1/2)

INT No.	Interrupt factors		Active level (Clearing standby)	CG interrupt mode control register
0	INT0	Interrupt Pin (PH0/AIN0/INT0)	High/Low Edge/Level Selectable	CGIMCGA
1	INT1	Interrupt Pin (PH1/AIN1/INT1)		
2	INT2	Interrupt Pin (PH2/AIN2/INT2)		
3	INT3	Interrupt Pin (PA0/TB0IN/INT3)		
4	INT4	Interrupt Pin (PA2/TB1IN/INT4)		CGIMCGB
5	INT5	Interrupt Pin (PE4/TB2IN//INT5)		
6	INTRX0	Serial reception (channel.0)		
7	INTTX0	Serial transmit (channel.0)		
8	INTRX1	Serial reception (channel.1)		
9	INTTX1	Serial transmit (channel.1)		
10	INTSSP0	Syncronous Serial Port 0		
11	INTSSP1	Syncronous Serial Port 1 (Note1)		
12	INTEMG0	PMD0 EMG interrupt (MPT0)		
13	INTEMG1	PMD1 EMG interrupt (MPT1) (Note1)		
14	INTSBI0	Serial Bus Interface 0 interrupt		
15	INTSBI1	Serial Bus Interface 1 interrupt (Note1)		
16	INTADPD0	ADC conversion triggered by PMD0 is finished		
17	INTRTC	Realtime clock interrupt	Low Edge	CGIMCGE
18	INTADPD1	ADC conversion triggered by PMD1 is finished		
19	INTRMCRX	Remote Controller reception interrupt	High Edge	CGIMCGE
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 (MPT0)		
29	INTPMD1	PMD1 PWM interrupt (MPT1) (Note1)		
30	INTCAP00	16bit TMRB0 input capture 0		
31	INTCAP01	16bit TMRB0 input capture 1		
32	INTCAP10	16bit TMRB1 input capture 0		
33	INTCAP11	16bit TMRB1 input capture 1		
34	INTCAP40	16bit TMRB4 input capture 0		
35	INTCAP41	16bit TMRB4 input capture 1		
36	INTCAP50	16bit TMRB5 input capture 0		
37	INTCAP51	16bit TMRB5 input capture 1		
38	INT6	Interrupt Pin (PE6/TB3IN/INT6) (Note1)	High/Low Edge/Level Selectable	CGIMCGB
39	INT7	Interrupt Pin (PE7/TB3OUT/INT7) (Note1)		
40	INTRX2	Serial reception (channel.2) (Note1)		
41	INTTX2	Serial transmit (channel.2) (Note1)		
42	INTADCP0	ADC conversion monitoring function interrupt 0		
43	INTADCP1	ADC conversion monitoring function interrupt 1		
44	INTRX4	Serial reception (channel.4) (Note1)		
45	INTTX4	Serial transmit (channel.4) (Note1)		

Table 7-2 List of Hardware Interrupt Factors (2/2)

No.	Interrupt factors		Active trigger (Clearing standby)	CG interrupt mode control register
46	INTTB20	16bit TMRB2 compare match detection 0/ Over flow		
47	INTTB21	16bit TMRB2 compare match detection 1		
48	INTTB30	16bit TMRB3 compare match detection 0/ Over flow		
49	INTTB31	16bit TMRB3 compare match detection 1		
50	INTCAP20	16bit TMRB2 input capture 0		
51	INTCAP21	16bit TMRB2 input capture 1		
52	INTCAP30	16bit TMRB3 input capture 0		
53	INTCAP31	16bit TMRB3 input capture 1		
54	INTADSFT	ADC conversion started by software is finished		
55	Reserved	Reserved		
56	INTADTMR	ADC conversion triggered by timer is finished		
57	Reserved	Reserved		
58	INT8	Interrupt Pin (PA7/TB4IN/INT8)	High/Low Edge/Level Selectable	CGIMCGC
59	INT9	Interrupt Pin (PD3/INT9) (Note1)		
60	INTA	Interrupt Pin (PJ6/AIN6/INTA) (Note1)		
61	INTB	Interrupt Pin (PJ7/AIN7/INTB) (Note1)		
62	INTENC0	Encoder input0 interrupt (Note1)		
63	INTENC1	Encoder input1 interrupt (Note1)		
64	INTRX3	Serial reception (channel.3) (Note1)		
65	INTTX3	Serial transmit (channel.3) (Note1)		
66	INTTB60	16bit TMRB6 compare match detection 0 / Over flow		
67	INTTB61	16bit TMRB6 compare match detection 1		
68	INTTB70	16bit TMRB7 compare match detection 0 / Over flow		
69	INTTB71	16bit TMRB7 compare match detection 1		
70	INTCAP60	16bit TMRB6 input capture 0		
71	INTCAP61	16bit TMRB6 input capture 1		
72	INTCAP70	16bit TMRB7 input capture 0		
73	INTCAP71	16bit TMRB7 input capture 1		
74	INTC	Interrupt Pin (PD0/ENCA0/TB5IN/INTC) (Note1)	High/Low Edge/Level Selectable	CGIMCGD
75	INTD	Interrupt Pin (PD2/ENCZ0/INTD) (Note1)		
76	INTE	Interrupt Pin (PN7/MT2IN/INTE) (Note1)		
77	INTF	Interrupt Pin (PL2/INTF)		
78	INTDMACERR	DMA transfer error		
79	INTDMACTC	DMA end of transfer		
80	INTMTTB00	16-bit MPT0 IGBT period/ compare match detection 0/ Over flow		
81	INTMTTB01	16-bit MPT0 IGBT trigger/ compare match detection 1		
82	INTMTTB10	16-bit MPT1 IGBT period/ compare match detection 0/ Over flow (Note1)		
83	INTMTTB11	16-bit MPT1 IGBT trigger/ compare match detection 1 (Note1)		
84	INTMTTB20	16-bit MPT2 IGBT period/ compare match detection 0/ Over flow (Note1)		
85	INTMTTB21	16-bit MPT2 IGBT trigger/ compare match detection 1 (Note1)		
86	INTMTCAP00	16-bit MPT0 input capture 0		
87	INTMTCAP01	16-bit MPT0 input capture 1		
88	INTMTCAP10	16-bit MPT1 input capture 0 (Note1)		
89	INTMTCAP11	16-bit MPT1 input capture 1 (Note1)		
90	INTMTCAP20	16-bit MPT2 input capture 0 (Note1)		
91	INTMTCAP21	16-bit MPT2 input capture 1 (Note1)		
92	INTMTEMG0	16-bit MPT0 IGBT EMG interrupt		
93	INTMTEMG1	16-bit MPT1 IGBT EMG interrupt (Note1)		
94	INTMTEMG2	16-bit MPT2 IGBT EMG interrupt (Note1)		

Note1 : for TMPM380

7.5.1.5 Active Level

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

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

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

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

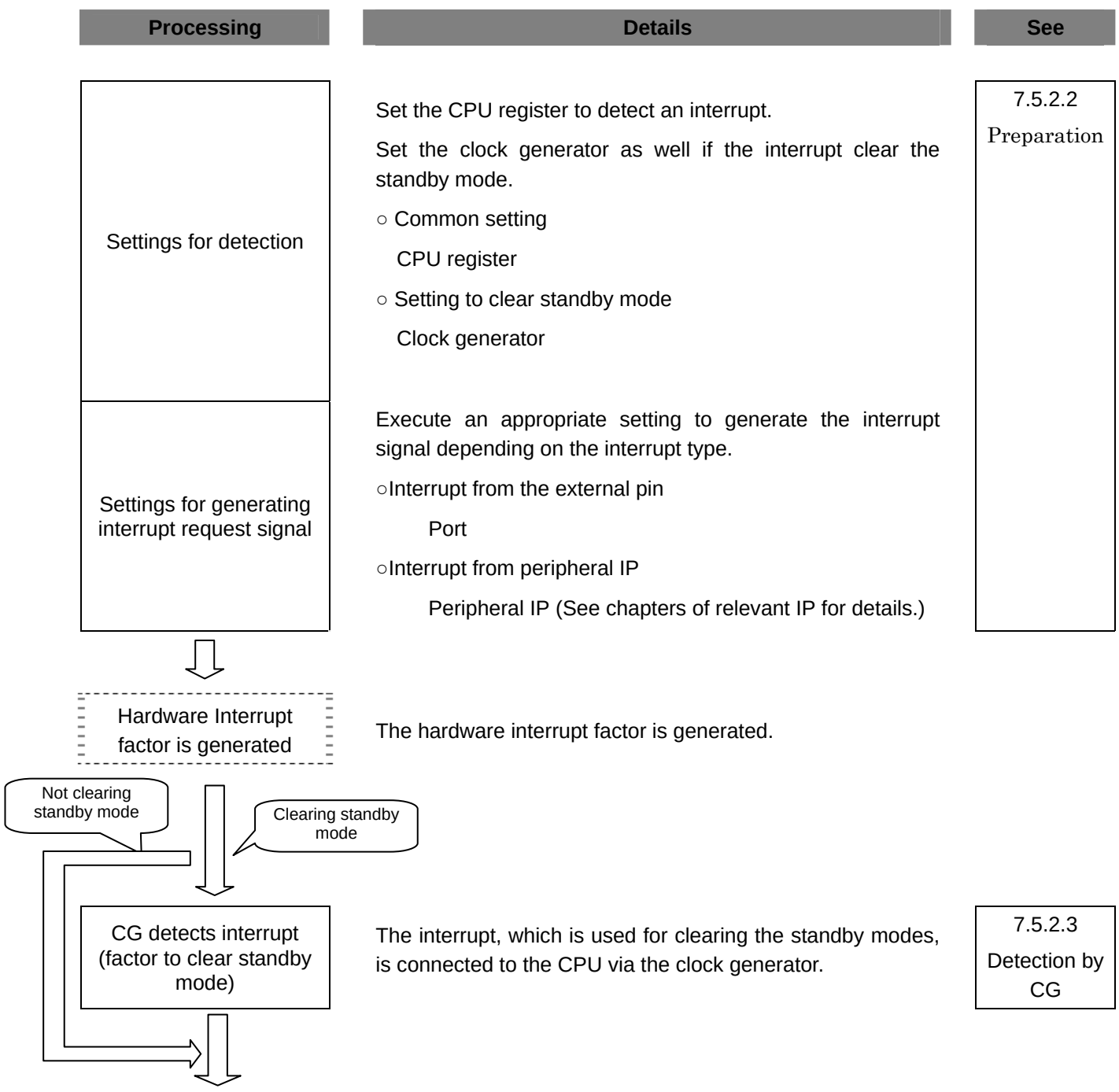
An interrupt from the external pin can be used without setting the clock generator in case it does not function as the standby clearing factor. However, inputting the “H” pulse or the “H” level signal is required so that the CPU can detect it as an interrupt factor.

7.5.2 Interrupt handling

7.5.2.1 Flowchart

The following shows how an interrupt is handled.

indicates hardware handling. indicates software handling.



Processing	Details	See
Detecting interrupt	The CPU detects the interrupt. In case several interrupt factor are detected, the interrupt factor with the highest priority is selected according to the priority order.	7.5.2.4 Detection by CPU
Handling interrupt	The CPU handles the interrupt. The CPU pushes resister contents to the stack before entering the interrupt service routine	7.5.2.5 CPU processing
Executing interrupt service routine	Program for the interrupt service routine. Clear the interrupt factor if needed.	7.5.2.6 Interrupt service routine
Returning to preceding program	Configure to return to the preceding program from the interrupt service routine.	

7.5.2.2 Preparation

When preparing for an interrupt, it is needed to pay attention to the order of configuration to avoid any unexpected interrupt on the way.

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

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

The following shows the order of interrupt handling and describe how to configure them.

(1) Disabling interrupt by CPU
(2) CPU registers setting
(3) Preconfiguration 1 (Interrupt from external pin)
(4) Preconfiguration 2 (interrupt from peripheral IP)
(5) Configuring the clock generator
(6) Enabling interrupt by CPU

(1) Disabling interrupt by CPU

To make the CPU for not accepting any interrupt, write “1” to the corresponding bit of the Interrupt Clear-Enable register. Each bit of the register, of which default setting is “0” with interrupt disabled, is assigned to single interrupt factor.

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

(Note) m: corresponding bit.

(2) CPU interrupt registers setting

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

Priority level 0 is the highest priority level.

●CPU register		
Interrupt Priority<m>	←	“Priority”

(Note) m: corresponding bit.

(3) Preconfiguration 1 (Interrupt from external pin)

Set “1” to the port function register of the corresponding pin. Setting PnFRx[m] allows the pin to be used as the function pin. Setting PnIE[m] allows the pin to be used as the input port.

● Port register		
PnFRx<PnmFRx>	←	“1”
PnIE<PnmIE>	←	“1”

(Note) n: port number
m: corresponding bit
x: function register number

(4) Preconfiguration 2 (interrupt from peripheral IP)

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

(5) Configuring the clock generator

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

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt. To clear corresponding interrupt request, write value corresponding to the interrupt to be used to the CGICRCG register. See 7.6.3.6 CG Interrupt Request Clear Register for each value.

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

●Clock generator register		
CGIMCGn<EMCGm>	←	Active level
CGICRCG<ICRCG>	←	Value corresponding to the interrupt to be used
CGIMCGn<INTmEN>	←	“1” (interrupt enabled)

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

(6) Enabling interrupt by CPU

Enable the interrupt by the CPU as shown below.

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

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

●CPU register		
Interrupt Clear-Pending<m>	←	“1”
Interrupt Set-Enable<m>	←	“1”

(Note) m: corresponding bit

7.5.2.3 Detection by Clock Generator

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

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

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

7.5.2.4 Detection by CPU

The CPU detects an interrupt factor with the highest priority.

7.5.2.5 CPU processing

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

7.5.2.6 Interrupt Service Routine

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

(1) Pushing during interrupt service routine

Common interrupt service routine is accompanied with the interrupt handling and the pushing of the register contents. The Cortex-M3 core automatically pushes the contents of PC, PSR, r0-r3, r12 and LR to the stack. No extra programming is required for them.

Push the contents of other registers if needed.

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

(2) Clearing interrupt factor

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

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

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

7.6 Exception/Interrupt-related registers

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

7.6.1 Register List

●NVIC Resisters	
SysTick Control and Status Resister	0xE000_E010
SysTick Reload Value Resister	0xE000_E014
SysTick Current Value Resister	0xE000_E018
SysTick Calibration Value Register	0xE000_E01C
Interrupt Set-Enable Register 1	0xE000_E100
Interrupt Set-Enable Register 2	0xE000_E104
Interrupt Set-Enable Register 3	0xE000_E108
Interrupt Clear-Enable Register 1	0xE000_E180
Interrupt Clear-Enable Register 2	0xE000_E184
Interrupt Clear-Enable Register 3	0xE000_E188
Interrupt Set-Pending Register 1	0xE000_E200
Interrupt Set-Pending Register 2	0xE000_E204
Interrupt Set-Pending Register 3	0xE000_E208
Interrupt Clear-Pending Register 1	0xE000_E280
Interrupt Clear-Pending Register 2	0xE000_E284
Interrupt Clear-Pending Register 3	0xE000_E288
Interrupt Priority Register	0xE000_E400 - 0xE000_E45C
Vector Table Offset Register	0xE000_ED08
System Handler Priority Register	0xE000_ED18,0xE000_ED1C,0xE000_ED20
System Handler Control and State Register	0xE000_ED24

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

7.6.2 NVIC Registers

7.6.2.1 SysTick Control and Status Register

	7	6	5	4	3	2	1	0
bit Symbol						CLK SOURCE	TICKINT	ENABLE
Read/Write	R					R/W	R/W	R/W
After reset	0					0	0	0
Function	“0” is read.					0: External reference clock 1: Core clock	0: Do not pend SysTick 1: Pend SysTick	0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							

	23	22	21	20	19	18	17	16
bit Symbol								COUNT FLAG
Read/Write	R							R/W
After reset	0							0
Function	“0” is read.							0: Timer not counted to 0 1: Timer counted to 0

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

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

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

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

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

7.6.2.2 SysTick Reload Value Register

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

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

(Note) In this product, the system timer counts based on a clock obtained by dividing the clock input from the X1 pin by 32.

7.6.2.3 SysTick Current Value Register

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

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

[Write] Writing to this register with any value clears it to 0. Clearing this register also clears the COUNTFLAG bit of the SysTick Control and Status Register.

7.6.2.4 SysTick Calibration Value Register

	7	6	5	4	3	2	1	0
bit Symbol	TENMS							
Read/Write	R							
After reset	1	1	0	0	0	1	0	0
Function	Calibration value (Note)							
	15	14	13	12	11	10	9	8
bit Symbol	TENMS							
Read/Write	R							
After reset	0	0	0	0	1	0	0	1
Function	Calibration value (Note)							
	23	22	21	20	19	18	17	16
bit Symbol	TENMS							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Calibration value (Note)							
	31	30	29	28	27	26	25	24
bit Symbol	NOREF	SKEW						
Read/Write	R	R	R					
After reset	0	0	0					
Function	0: Reference clock provided 1: No reference clock	0: Calibration value is 10 ms. 1: Calibration value is not 10 ms.	"0" is read.					

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

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

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

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

7.6.2.5 Interrupt Set-Enable Register 1

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 7 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 6 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 5 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 4 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 3 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 2 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 1 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 0 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	15	14	13	12	11	10	9	8
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 15 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 14 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 13 [Write] 1: Enable [Read] 0 Disabled 1: Enabled	Interrupt number 12 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 11 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 10 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 9 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 8 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	23	22	21	20	19	18	17	16
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 23 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 22 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 21 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 20 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 19 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 18 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 17 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number16 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	31	30	29	28	27	26	25	24
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 31 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 30 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 29 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 28 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 27 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 26 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 25 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 24 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

<bit31:0> <SETENA> Use these bits to enable interrupts or determine which interrupts are currently enabled.

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

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

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors**.**

7.6.2.6 Interrupt Set-Enable Register 2

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

<bit31:0> <SETENA> Use these bits to enable interrupts or determine which interrupts are currently enabled.

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

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

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors**.**

7.6.2.7 Interrupt Set-Enable Register 3

	7	6	5	4	3	2	1	0
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 71 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 70 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 69 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 68 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 67 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 66 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 65 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 64 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	15	14	13	12	11	10	9	8
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 79 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 78 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 77 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 76 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 75 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 74 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 73 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 72 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	23	22	21	20	19	18	17	16
bit Symbol	SETENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 87 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 86 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 85 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 84 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 83 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 82 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 81 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 80 [Write] 1: Enable [Read] 0: Disabled 1: Enabled
	31	30	29	28	27	26	25	24
bit Symbol		SETENA						
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Interrupt number 94 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 93 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 92 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 91 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 90 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 89 [Write] 1: Enable [Read] 0: Disabled 1: Enabled	Interrupt number 88 [Write] 1: Enable [Read] 0: Disabled 1: Enabled

<bit30:0> <SETENA> Use these bits to enable interrupts or determine which interrupts are currently enabled.

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

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

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors**.**

7.6.2.8 Interrupt Clear-Enable Register 1

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 7 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 6 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 5 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 4 [Write] 1: Disable [read] 0: Disabled 1: Enabled	Interrupt number 3 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 2 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 1 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 0 [Write] 1: Disable [Read] 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 15 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 14 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 13 [Write] 1: Disable [Read] 0: Disabled 1 Enabled	Interrupt number 12 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 11 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 10 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 9 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 8 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	23	22	21	20	19	18	17	16
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 23 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 22 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 21 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 20 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 19 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 18 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 17 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 16 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	31	30	29	28	27	26	25	24
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 31 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 30 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 29 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 28 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 27 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 26 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 25 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 24 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

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

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

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

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors**.**

7.6.2.9 Interrupt Clear-Enable Register 2

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 39 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 38 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 37 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 36 [Write] 1: Disable [read] 0: Disabled 1: Enabled	Interrupt number 35 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 34 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 33 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 32 [Write] 1: Disable [Read] 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 47 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 46 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 45 [Write] 1: Disable [Read] 0: Disabled 1 Enabled	Interrupt number 44 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 43 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 42 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 41 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 40 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	23	22	21	20	19	18	17	16
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 55 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 54 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 53 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 52 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 51 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 50 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 49 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 48 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	31	30	29	28	27	26	25	24
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 63 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 62 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 61 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 60 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 59 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 58 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 57 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 56 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

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

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

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

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors**.**

.

7.6.2.10 Interrupt Clear-Enable Register 3

	7	6	5	4	3	2	1	0
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 71 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 70 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 69 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 68 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 67 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 66 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 65 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 64 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	15	14	13	12	11	10	9	8
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 79 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 78 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 77 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 76 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 75 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 74 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 73 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 72 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	23	22	21	20	19	18	17	16
bit Symbol	CLRENA							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt number 87 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 86 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 85 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 84 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 83 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 82 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 81 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 80 [Write] 1: Disable [Read] 0: Disabled 1: Enabled
	31	30	29	28	27	26	25	24
bit Symbol		CLRENA						
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Interrupt number 94 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 93 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 92 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 91 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 90 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 89 [Write] 1: Disable [Read] 0: Disabled 1: Enabled	Interrupt number 88 [Write] 1: Disable [Read] 0: Disabled 1: Enabled

<bit30:0> <CLRENA> Use these bits to disable or determine which interrupts are currently disabled.

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

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

0 = Disabled
1 = Enabled

(Note) For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4 List of Interrupt Factors**.**

7.6.2.11 Interrupt Set-Pending Register 1

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 7 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 6 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 5 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 4 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 3 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 2 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 1 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 0 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	15	14	13	12	11	10	9	8
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 15 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 14 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 13 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 12 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 11 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number10 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 9 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 8 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	23	22	21	20	19	18	17	16
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 23 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 22 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 21 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 20 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 19 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 18 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 17 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 16 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	31	30	29	28	27	26	25	24
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 31 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 30 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 29 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 28 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 27 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 26 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 25 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 24 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit31:0>	<SETPEND>	Use these bits to force interrupts into the pending state or determine which interrupts are currently pending. Writing “1” to a bit in this register pends the corresponding interrupt. However, writing “1” has no effect on an interrupt that is already pending or is disabled. Writing “0” has no effect. Reading a bit in this register returns the current state of the corresponding interrupt as shown below. 0 = Not pending 1 = Pending Each bit in this register can be cleared by writing “1” to the corresponding bit in the Interrupt Clear-Pending Register.

(Note) **For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4** List of Interrupt Factors**.**

7.6.2.12 Interrupt Set-Pending Register 2

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 39 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 37 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 36 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	15	14	13	12	11	10	9	8
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 47 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 46 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 45 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 44 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 43 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number42 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 41 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number40 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	23	22	21	20	19	18	17	16
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 55 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 54 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 53 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 52 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 51 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 50 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 49 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 48 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	31	30	29	28	27	26	25	24
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 63 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 62 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 61 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 60 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 59 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 58 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 57 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 56 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit31:0>	<SETPEND>	Use these bits to force interrupts into the pending state or determine which interrupts are currently pending. Writing “1” to a bit in this register pends the corresponding interrupt. However, writing “1” has no effect on an interrupt that is already pending or is disabled. Writing “0” has no effect. Reading a bit in this register returns the current state of the corresponding interrupt as shown below. 0 = Not pending 1 = Pending Each bit in this register can be cleared by writing “1” to the corresponding bit in the Interrupt Clear-Pending Register.

(Note) **For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4** List of Interrupt Factors**.**

7.6.2.13 Interrupt Set-Pending Register 3

	7	6	5	4	3	2	1	0
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 71 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 70 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 69 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 68 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 67 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 66 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 65 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 64 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	15	14	13	12	11	10	9	8
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 79 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 78 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 77 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 76 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 75 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 74 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 73 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 72 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	23	22	21	20	19	18	17	16
bit Symbol	SETPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 87 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 86 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 85 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 84 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 83 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 82 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 81 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 80 [Write] 1: Pend [Read] 0: Not pending 1: Pending
	31	30	29	28	27	26	25	24
bit Symbol		SETPEND						
Read/Write	R	R/W						
After reset	0	Undefined						
Function	“0” is read.	Interrupt number 94 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 93 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 92 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 91 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 90 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 89 [Write] 1: Pend [Read] 0: Not pending 1: Pending	Interrupt number 88 [Write] 1: Pend [Read] 0: Not pending 1: Pending

<bit30:0>	<SETPEND>	Use these bits to force interrupts into the pending state or determine which interrupts are currently pending. Writing “1” to a bit in this register pends the corresponding interrupt. However, writing “1” has no effect on an interrupt that is already pending or is disabled. Writing “0” has no effect. Reading a bit in this register returns the current state of the corresponding interrupt as shown below. 0 = Not pending 1 = Pending Each bit in this register can be cleared by writing “1” to the corresponding bit in the Interrupt Clear-Pending Register.

(Note) **For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4** List of Interrupt Factors**.**

7.6.2.14 Interrupt Clear-Pending Register 1

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

<bit31:0>	<CLRPEND>	Use these bits to clear pending interrupts or determine which interrupts are currently pending. Writing “1” to a bit in this register clears the corresponding pending interrupt. However, writing “1” has no effect on an interrupt that is already being serviced. Writing “0” has no effect. Reading a bit in this register returns the current state of the corresponding interrupt as shown below. 0 = Not pending 1 = Pending

(Note) **For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4** List of Interrupt Factors**.**

7.6.2.15 Interrupt Clear-Pending Register 2

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number39 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 38 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 37 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 36 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 35 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 34 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 33 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 32 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 47 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 46 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 45 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 44 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 43 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 42 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 41 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 40 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	23	22	21	20	19	18	17	16
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 55 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 54 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 53 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 52 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 51 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 50 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 49 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 48 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	31	30	29	28	27	26	25	24
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 63 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 62 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 61 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 60 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 59 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number58 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 57 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 56 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

<bit31:0>	<CLRPEND>	Use these bits to clear pending interrupts or determine which interrupts are currently pending. Writing “1” to a bit in this register clears the corresponding pending interrupt. However, writing “1” has no effect on an interrupt that is already being serviced. Writing “0” has no effect. Reading a bit in this register returns the current state of the corresponding interrupt as shown below. 0 = Not pending 1 = Pending

(Note) **For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4** List of Interrupt Factors**.**

7.6.2.16 Interrupt Clear-Pending Register 3

	7	6	5	4	3	2	1	0
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 71 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 70 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 69 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 68 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 67 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 66 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 65 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 64 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	15	14	13	12	11	10	9	8
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 79 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 78 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 77 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 76 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 75 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 74 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 73 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 72 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	23	22	21	20	19	18	17	16
bit Symbol	CLRPEND							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt number 87 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 86 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 85 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 84 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 83 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 82 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 81 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 80 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
	31	30	29	28	27	26	25	24
bit Symbol		CLRPEND						
Read/Write	R	R/W						
After reset	0	Undefined						
Function	“0” is read.	Interrupt number 94 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 93 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 92 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 91 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 90 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 89 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending	Interrupt number 88 [Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending

<bit30:0>	<CLRPEND>	Use these bits to clear pending interrupts or determine which interrupts are currently pending. Writing “1” to a bit in this register clears the corresponding pending interrupt. However, writing “1” has no effect on an interrupt that is already being serviced. Writing “0” has no effect. Reading a bit in this register returns the current state of the corresponding interrupt as shown below. 0 = Not pending 1 = Pending

(Note) **For descriptions of interrupts and interrupt numbers, see Section 7.5.1.4** List of Interrupt Factors**.**

7.6.2.17 Interrupt Priority Registers

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

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

	31 24	23 16	15 8	7 0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000_E434	PRI_55	PRI_54	PRI_53	PRI_52
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000_E44C	PRI_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_81	PRI_80	PRI_89	PRI_88
0xE000_E45C	-	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.

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

<bit7:5> <PRI_0> Priority of interrupt number 0

<bit15:13> <PRI_1> Priority of interrupt number 1

<bit23:21> <PRI_2> Priority of interrupt number 2

<bit31:29> <PRI_3> Priority of interrupt number 3

7.6.2.18 Vector Table Offset Register

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

<bit28:7> <TBLOFF> Set the offset value from the top of the space specified in TBLBASE.
The offset must be aligned based on the number of exceptions in the table. This means that the minimum alignment is 32 words that you can use for up to 16 interrupts. For more interrupts, it is necessary to adjust the alignment by rounding up to the next power of two.

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

7.6.2.19 System Handler Priority Registers

System Handler Priority Registers have eight bits per each exception.

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

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

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

The following shows the fields of the System Handler Priority Registers for Memory Management, Bus Fault and Usage Fault. The System Handler Priority Registers for all other exceptions have the identical fields. Unused bits return “0” when read, and writing to unused bits has no effect.

	7	6	5	4	3	2	1	0
bit Symbol	PRI_4							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Memory Management			“0” is read.				
	15	14	13	12	11	10	9	8
bit Symbol	PRI_5							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Bus Fault			“0” is read.				
	23	22	21	20	19	18	17	16
bit Symbol	PRI_6							
Read/Write	R/W			R				
After reset	0			0				
Function	Priority of Usage Fault			“0” is read.				
	31	30	29	28	27	26	25	24
bit Symbol	PRI_7							
Read/Write	R/W			R				
After reset	0			0				
Function	Reserved			“0” is read.				

7.6.2.20 System Handler Control and State Register

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

<bit0>	<MEMFAULTACT>	Reads as “1” if Memory Management is active.
<bit1>	<BUSFAULTACT>	Reads as “1” if Bus Fault is active.
<bit3>	<USGFALTACT>	Reads as “1” if Usage Fault is active.
<bit7>	<SVCALLACT>	Reads as “1” if SVCall is active.
<bit8>	<MONITORACT>	Reads as “1” if Debug Monitor is active.
<bit10>	<PENDSVACT>	Reads as “1” if PendSV is active.
<bit11>	<SYSTICKACT>	Reads as “1” if SysTick is active.
<bit12>	<USGFAULTPENDED>	Reads as “1” if Usage Fault is pended.
<bit13>	<MEMFAULTPENDED>	Reads as “1” if Memory Management is pended.
<bit14>	<BUSFAULTPENDED>	Reads as “1” if Bus Fault is pended.
<bit15>	<SVCALLPENDED>	Reads as “1” if SVCall is pended.
<bit16>	<MEMFAULTENA>	Set to “0” to disable or “1” to enable Memory Management.
<bit17>	<BUSFAULTENA>	Set to “0” to disable or “1” to enable Bus Fault.
<bit18>	<USGFAULTENA>	Set to “0” to disable or “1” to enable Usage Fault.

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

7.6.3 Clock Generator Registers

7.6.3.1 CG Interrupt Mode Control Register A

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

CGIMCGA
0x4004_0220

	7	6	5	4	3	2	1	0
bit Symbol		EMCG02	EMCG01	EMCG00	EMST01	EMST00		INT0EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT0 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT0 standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT0 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG12	EMCG11	EMCG10	EMST11	EMST10		INT1EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT1 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT1 standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT1 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG22	EMCG21	EMCG20	EMST21	EMST20		INT2EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT2 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT2 standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT2 clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG32	EMCG31	EMCG30	EMST31	EMST30		INT3EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT3 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT3 standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT3 clear input 0: Disable 1: Enable

(Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

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

7.6.3.2 CG Interrupt Mode Control Register B

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

CGIMCGB

0x4004_0224

	7	6	5	4	3	2	1	0
bit Symbol		EMCG42	EMCG41	EMCG40	EMST41	EMST40		INT4EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT4 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT4 standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT4 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG52	EMCG51	EMCG50	EMST51	EMST50		INT5EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT5 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT5 standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT5 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG62	EMCG61	EMCG60	EMST61	EMST60		INT6EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT6 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT6 standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT6 clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG72	EMCG71	EMCG70	EMST71	EMST70		INT7EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT7 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT7 standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT7 clear input 0: Disable 1: Enable

(Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

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

(Note3) INT6 to INT7 is for TMPM380 Only. Write "0" to Bit[24][16] and "010" to Bit[30:28][22:20] for TMPM382.

7.6.3.3 CG Interrupt Mode Control Register C

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

CGIMCGC
0x4004_0228

	7	6	5	4	3	2	1	0
bit Symbol		EMCG82	EMCG81	EMCG80	EMST81	EMST80		INT8EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT8 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT8 standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT8 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG92	EMCG91	EMCG90	EMST91	EMST90		INT9EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INT9 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT9 standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INT9 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCGA2	EMCGA1	EMCGA0	EMSTA1	EMSTA0		INTAEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTA standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTA standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INTA clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCGB2	EMCGB1	EMCGB0	EMSTB1	EMSTB0		INTBEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTB standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTB standby clear request 00: ー 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INTB clear input 0: Disable 1: Enable

(Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

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

(Note3) INT9 to INTB is for TMPM380 Only. Write "0" to Bit[24][16][8] and "010" to Bit[30:28][22:20][14:12] for TMPM382.

7.6.3.4 CG Interrupt Mode Control Register D

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

CGIMCGD

0x4004_022C

	7	6	5	4	3	2	1	0
bit Symbol		EMCGC2	EMCGC1	EMCGC0	EMSTC1	EMSTC0		INTCEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTC standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTC standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INTC clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol		EMCGD2	EMCGD1	EMCGD0	EMSTD1	EMSTD0		INTDEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTD standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTD standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INTD clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol		EMCGE2	EMCGE1	EMCGE0	EMSTE1	EMSTE0		INTEEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTE standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTE standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		"0" is read	INTE clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol		EMCGF2	EMCGF1	EMCGF0	EMSTF1	EMSTF0		INTFEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTF standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTF standby clear request 00: ― 01: Rising edge 10: Falling edge 11: Both edges		Reads as undefined.	INTF clear input 0: Disable 1: Enable

(Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

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

(Note3) INTC to INTE is for TMPM380 Only. Write "0" to Bit[16][8][0] and "010" to Bit[22:20][14:12][6:4] for TMPM382.

7.6.3.5 CG Interrupt Mode Control Register E

This resister set the clearing standby request active level of interrupt INTRTC, INTRMCRX.

CGIMCGE

0x4004_0230

	7	6	5	4	3	2	1	0
bit Symbol		EMCGR TC2	EMCGR TC1	EMCGR TC0	EMSRT TC1	EMSTR TC0		INTRTC EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTRTC standby clear request. Write "010" 010: Falling edge			Active state of INTRTC standby clear request 00: ー 01: ー 10: Falling edge 11: ー		Reads as undefined.	INTRTC clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCGR MCRX2	EMCGR MCRX1	EMCGR MCRX0	EMSTR MCRX1	EMSTR MCRX0		INTRMC RXEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	Undefined	0
Function	"0" is read	Active state setting of INTRMCRX standby clear request. Write "011" 011: Rising edge			Active state of INTRMCRX standby clear request 00: ー 01: Rising edge 10:ー 11: ー		Reads as undefined.	INTRMCRX clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read	Write any value.			"00" is read.		"0" is read.	Write "0".

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read	Write any value.			"00" is read.		"0" is read.	Write "0".

(Note1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

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

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

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

1) If the external intrrupt request pin shared with other general ports, enable the port to receive the interrupt.
2) Set conditions such as active state upon initialization.
3) Clear interrupt requests.
4) Enable interrupts.

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

(Note 3) For clearing the Standby modes , 18 interrupt factors (INT0 to INTF,INTRTC,INTRMCRX) are available. CGIMCGA~CGIMCGE Resisters in CG are used for selecting edge/level of active state and which aforementioned factors are used for clearing the standby modes.

(Note 4) In case the standby mode clear is not required, interrupt factors (INT0 to INTF,INTRTC,INTRMCRX) can be used as a normal interrupt without setting CGIMCGA~CGIMCGE Resisters in CG.

7.6.3.6 CG Interrupt Request Clear Register

This resister clear the Interrupt request from INT0~INTF,INTRTC, INTRMCRX.

CGICRCG

	7	6	5	4	3	2	1	0
bit Symbol				ICRCG4	ICRCG3	ICRCG2	ICRCG1	ICRCG0
Read/Write	R			W				
After reset	0			0	0	0	0	0
Function	"0" is read.			Clear interrupt requests 0_0000:INT0 0_1000:INT8 1_0000:INTRTC 0_0001:INT1 0_1001:INT9(Note1) 1_0001:INTRMCRX 0_0010:INT2 0_1010:INTA(Note1) 0_0011:INT3 0_1011:INTB(Note1) 0_0100:INT4 0_1100:INTC(Note1) 0_0101:INT5 0_1101:INTD(Note1) 0_0110:INT6(Note1) 0_1110:INTE(Note1) 0_0111:INT7(Note1) 0_1111:INTF 1_0010～1_1111: setting prohibited * "0" is read.				

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

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

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

(Note1) For TMPM380.

7.6.3.7 NMI Flag Register

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

CGNMIFLG

	7	6	5	4	3	2	1	0
bit Symbol						NMIFLG2	-	NMIFLG0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.					NMI factor generation flag 0: not applicable 1: generated from Voltage level detection	-	NMI factor generation flag 0:not applicable 1: generated from WDT

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

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

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

(Note) <NMIFLG2:0> are cleared to "0" when they are read.

7.6.3.8 Reset Flag Register

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

CGRSTFLG

	7	6	5	4	3	2	1	0
bit Symbol			OFDRSTF	DBGRSTF		WDTRSTF	PINRSTF	PONRSTF
Read/Write	R		R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	“0” is read.		OFD reset flag 0:“0” is written 1:Reset from OFD	Debug reset flag 0: “0” is written 1:Reset from SYSRSTRQ	Write “0”.	WDT reset flag 0:“0” is written 1:Reset from WDT	RESET pin flag 0: “0” is written 1:Reset from RESET pin	Power On Reset flag 0: “0” is written 1:Reset from Power On Reset

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

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

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

(Note1) The TMPM380 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. This bit is not set by the second and subsequent resets .

(Note2) SYSRSTRQ is occurred by set to SYSRESETREQ bit in NVIC.

8 Input/Output Ports

8.1 Port registers

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

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

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

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

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

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

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

8.2 Port Functions

8.2.1 Port Status in Stop Mode

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

If PxIE or PxCR is enabled with <DRVE>=1, input or output is enabled respectively in STOP mode.

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

The differences are summarized in the table shown below.

Table 8-1 Port status in Stop Mode

	Pin Name	I/O	<DRVE>=0	<DRVE>=1
control pins	$\overline{\text{RESET}}$, MODE	Input only	○	○
Oscillator	X1,XT1	Input only	×	×
	X2, XT2	Output only	“H” level output	“H” level output
Ports	PAn to PPn	Input	×	Depends on PxIE<n>.
		Output	×	Depends on PxCR<n>.
Debag Interface	TMS/SWDIO/TDO/SWV	Input	×	Depends on PxIE<n>
		Output	●	●
External Interrupts	INT0 to INTF	Input	Depends on PxIE<n>.	Depends on PxIE<n>.
SSP	SPnCLK/SPnFSS/SPnDO	Output	×	●
MPT(IGBT)	$\overline{\text{GEMGn}}$	Input	×	Depends on PxIE<n>.
	MTnOUTxx	Output	●	●
MPT(PMD)	$\overline{\text{EMGn}}$	Input	×	Depends on PxIE<n>.
	UOn/VOn/WOn/XOn/YOn/ZOn	Output	●	●
Others		Input	×	Depends on PxIE<n>.
		Output	×	Depends on PxCR<n>.

○ : Input or output enabled

× : Input or output disabled.

● : **Depend on PxCR<n> and Enabled when data is valid.**

n : Bit number

8.2.2 Port A (PA0 to PA7)

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

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

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

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

Port A register

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

Port A control register

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

Port A function register 1

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

Port A function register 2

PAFR2 (0x4000_000C)

	7	6	5	4	3	2	1	0
Bit Symbol	PA7F2	PA6F2	PA5F2	PA4F2	PA3F2	PA2F2	PA1F2	PA0F2
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1 INT8	0:PORT 1:TB6IN	0:PORT 1:TB6OUT	0:PORT 1:CTS1	0:PORT 1:RXIN	0:PORT 1:INT4	0:PORT 1:SCOUT	0:PORT 1:INT3

Port A open drain Control register

PAOD (0x4000_ 0028)

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

Port A pull-up control register

PAPUP (0x4000_002C)

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

Port A pull-down control register

PAPDN (0x4000_0030)

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

Port A input enable control register

PAIE (0x4000_0038)

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

8.2.3 Port B (PB0 to PB7)

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

Reset initializes PB3, PB4, PB5, PB6 and PB7 to perform debug communication function.When PB3 functions as the TMS or SWDIO, input, output and pull-up are enabled. When PB4 functions as the TCK or SWCLK, input, pull-down are enabled. When PB5 functions as the TDO or SWV, output is enabled. When PB6 functions TDI, input, pull-up are enabled. When PB7 functions as $\overline{\text{TRST}}$ input, pull-up is enabled. PB0, PB1, PB2 perform as the general-purpose ports with input, output and pull-up disabled.

(Note 1)	**The default setting for PB3 is function port. Input, output, and pull-up are enabled.**
(Note 2)	**The default setting for PB4 is function port. Input, and pull-down are enabled.**
(Note 3)	**The default setting for PB5 is function port. Output is enabled.**
(Note 4)	**The default setting for PB6 and PB7 are function port. Input and pull-up are enabled.**
(Note 5)	**If PB3 and PB5 are configured to function port for debug function, outputs are enabled even during Stop mode regardless of the CGSTBYCR<DRVE> bit setting.**

Port B register

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

Port B control register

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

Port B function register 1

PBFR1
(0x4000_0048)

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

Port B open drain Control register

PBOD
(0x4000_ 0068)

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

Port B pull-up control register

PBPUP
(0x4000_006C)

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

Port B pull-down control register

PBPDN
(0x4000_0070)

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

Port B input enable control register

PBIE
(0x4000_0078)

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

8.2.4 Port C (PC0 to PC7)

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

.

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

Port C register

PCDATA (0x4000_0080)

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

Port C control register

PCCR (0x4000_0084)

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

Port C function register 1

PCFR1 (0x4000_0088)

	7	6	5	4	3	2	1	0
Bit Symbol	-	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	”0” is read	0:port 1:$\overline{\text{EMG0}}$	0:port 1:ZO0	0:port 1:WO0	0:port 1:YO0	0:port 1:VO0	0:port 1:XO0	0:port 1:UO0

Port C function register 2

PCFR2 (0x4000_008C)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7F2	PC6F2	PC5F2	PC4F2	PC3F2	PC2F2	PC1F2	PC0F2
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:port 1:MT0IN	0:port 1:$\overline{\text{GEMG0}}$	0:port 1:MTOUT10	0:port 1:MTOUT00	0:port 1:SP0FSS	0:port 1:SP0CLK	0:port 1:SP0DI	0:port 1:SP0DO

Port C function register 3

PCFR3 (0x4000_0090)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	PC5F3	PC4F3	-	PC2F3	PC1F3	PC0F3
Read/Write	R		R/W		R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read		0:port 1:MTTB0IN	0:port 1:MTTB0OUT	“0” is read	0:port 1:SCK0	0:port 1:SI0/SCL0	0:port 1:SO0/SDA0

Port C function register 4

PCFR4 (0x4000_0094)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7F4	PC6F4	PC5F4	-	-	-	-	-
Read/Write	R/W			R				
After reset	0	0	0	0	0	0	0	0
Function	0:port 1:RX4	0:port 1:TX4	0:port 1:SCLK4	“0” is read				

Port C function register 5

PCFR5 (0x4000_0098)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	PC5F4	-	-	-	-	-
Read/Write	R		R/W	R				
After reset	0	0	0	0	0	0	0	0
Function	“0” is read		0:port 1:CTS4	“0” is read				

Port C open drain Control register

PCOD (0x4000_ 00A8)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7OD	PC6OD	PC5OD	PC4OD	PC3OD	PC2OD	PC1OD	PC0OD
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:CMOS 1:open drain							

Port C pull-up control register

PCPUP (0x4000_00AC)

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

Port C pull-down control register

PCPDN (0x4000_00B0

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

Port C input enable control register

PCIE (0x4000_00B8)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE	PC0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0:disabled 1:enabled							

8.2.5 Port D (PD0 to PD6)

(Important)
TMPM382 (64 pin version) does not implement port D (PD0 to PD6).
Please do not use these functions if you use this product.

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

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

To use the external interrupt input for releasing STOP mode, select this function in the PDFR1,PDFR3 register and enable input in the PDIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

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

Port D register

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

Port D control register

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

Port D function register 1

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

Port D function register 2

PDFR2 (0x4000_00CC)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	PD4F2	—	—	PD1F2	PD0F2
Read/Write	R			R/W	R		R/W	R/W
After reset	0			0	0		0	0
Function	“0” is read.			0:PORT 1:CTS2	“0” is read.		0:PORT 1:TB5OUT	0:PORT 1: TB5IN

Port D function register 3

PDFR3 (0x4000_00D0)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	PD2F1	—	PD0F1
Read/Write	R					R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.					0:PORT 1:INTD	“0” is read.	0:PORT 1:INTC

Port D open drain control register

PDOD (0x4000_ 00E8)

	7	6	5	4	3	2	1	0
Bit Symbol	—	PD6OD	PD5OD	PD4OD	PD3OD	PD2OD	PD1OD	PD0OD
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	0:CMOS 1: open drain						

Port D pull-up control register

PDPUP (0x4000_00EC)

	7	6	5	4	3	2	1	0
Bit Symbol	-	PD6UP	PD5UP	PD4UP	PD3UP	PD2UP	PD1UP	PD0UP
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Pull-up 0:disabled 1:enabled						

Port D pull-down control register

PDPDN (0x4000_00F0)

	7	6	5	4	3	2	1	0
Bit Symbol	-	PD6DN	PD5DN	PD4DN	PD3DN	PD2DN	PD1DN	PD0DN
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Pull-down 0:disabled 1:enabled						

Port D input enable control register

PDIE (0x4000_00F8)

	7	6	5	4	3	2	1	0
Bit Symbol	-	PD6IE	PD5IE	PD4IE	PD3IE	PD2IE	PD1IE	PD0IE
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Input 0:disabled 1:enabled						

8.2.6 Port E (PE0 to PE7)

(Important)
TMPM382 (64 pin version) does not implement PE6 and PE7.
Please do not use these functions if you use this product.

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

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

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

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

Port E register

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

Port E control register

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

Port E functions register 1

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

Port E functions register 2

PEFR2 (0x4000_010C)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7F2	PE6F2	−	PE4F2	−	PE2F2	−	−
Read/Write	R/W	R/W	R	R/W	R	R/W	R	
After reset	0	0	0	0	0	0	0	
Function	0:PORT 1:INT7.	0:PORT 1:INT6	“0” is read.	0:PORT 1:INT5	“0” is read.	0:PORT 1:CTS0	“0” is read.	

Port E open drain control register

PEOD (0x4000_ 0128)

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

Port E pull-up control register

PEPUP (0x4000_012C)

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

Port E pull-down control register

PEPDN (0x4000_0130)

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

Port E input enable control register

PEIE (0x4000_0138)

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

8.2.7 Port F (PF0 to PF4)

(Important)
TMPM382 (64 pin version) does not implement PF2,PF3 and PF4.
Please do not use these functions if you use this product.

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

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

Port F register

PFDATA (0x4000_0140)

	7	6	5	4	3	2	1	0
Bit Symbol	−	−	−	PF4	PF3	PF2	PF1	PF0
Read/Write	R			R/W				
After reset	“0” is read.			“0”				

Port F control register

PFCR (0x4000_0144)

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

Port F function register 1

PFFR1 (0x4000_0148)

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

Port F function register 2

PFFR2 (0x4000_014C)

	7	6	5	4	3	2	1	0
Bit Symbol	−	−	−	PF4F2	PF3F2	PF2F2	PF2F2	−
Read/Write	R			R/W				R
After reset	0			0	0	0	0	0
Function	“0” is read.			0:PORT 1:RXD3	0:PORT 1:TXD3	0:PORT 1:SCLK3	.0:PORT 1:$\overline{\text{ALARM}}$	“0” is read

Port F function register 3

PFFR3 (0x4000_0150)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	–	–	PF2F3	–	–
Read/Write	R					R/W	R	
After reset	0					0	0	
Function	“0” is read.					0:PORT 1:CTS3	“0” is read.	

Port F open drain control register

PFOD (0x4000_0168)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	PF4OD	PF3OD	PF2OD	PF1OD	PF0OD
Read/Write	R			R/W				
After reset	0			0	0	0	0	0
Function	“0” is read			0:CMOS 1:Open drain				

Port F pull-up control register

PFPUP (0x4000_016C)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	PF4UP	PF3UP	PF2UP	PF1UP	PF0UP
Read/Write	R			R/W				
After reset	0			0	0	0	0	0
Function	“0” is read			Pull-up 0:disabled 1:enabled				

Port F pull-down control register

PFPDN (0x4000_0170)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	PF4DN	PF3DN	PF2DN	PF1DN	PF0DN
Read/Write	R			R/W				
After reset	0			0	0	0	0	0
Function	“0” is read			Pull-down 0:disabled 1:enabled				

Port F input enable control register

PFIE (0x4000_0178)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
Read/Write	R			R/W				
After reset	0			0	0	0	0	0
Function	“0” is read			Input 0:disabled 1:enabled				

8.2.8 Port G (PG0 to PG7)

(Important)
TMPM382 (64 pin version) does not implement port G (PG0 to PG7).
Please do not use these functions if you use this product.

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

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

Port G register

PGDATA (0x4000_0180)

	7	6	5	4	3	2	1	0
Bit Symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
Read/Write	R/W							
After reset	“0”							

Port G control register

PGCR (0x4000_0184)

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

Port G function register 1

PGFR1 (0x4000_0188)

	7	6	5	4	3	2	1	0
Bit Symbol	-	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	”0” is read	0:PORT 1: $\overline{\text{EMG1}}$	0:PORT 1: ZO1	0:PORT 1: WO1	0:PORT 1: YO1	0:PORT 1: VO1	0:PORT 1: XO1	0:PORT 1: UO1

Port G function register 2

PGFR2 (0x4000_018C)

	7	6	5	4	3	2	1	0
Bit Symbol	PG7F2	PG6F2	PG5F2	PG4F2	-	-	-	-
Read/Write	R/W				R			
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1: MT1IN	0:PORT 1: $\overline{\text{GEMG1}}$	0:PORT 1: MTOUT01	0:PORT 1: MTOUT00	“0” is read			

Port G function register 3

PGFR3 (0x4000_0190)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	PG5F1	PG4F1	-	PG2F1	PG1F1	PG0F1
Read/Write	R		R/W		R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read		0:PORT 1: MTTB1IN	0:PORT 1:MTTB1 OUT	"0" is read	0:PORT 1: SCK1	0:PORT 1:SI1/SCL 1	0:PORT 1: SO1/SDA 1

Port G open drain control register

PGOD (0x4000_01A8)

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

Port G pull-up control register

PGPUP (0x4000_01AC)

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

Port G pull-down control register

PGPDN (0x4000_01B0)

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

Port G input enable control register

PGIE (0x4000_01B8)

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

8.2.9 Port H (PH0 to PH7)

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

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

To use the external interrupt input for releasing STOP mode, select this function in the PHFR1 register and enable input in the PHIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

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

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

Port H register

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

Port H control register

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

Port H function register 1

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

Port H open drain control register

PHOD (0x4000_ 01E8)

	7	6	5	4	3	2	1	0
Bit Symbol	PH7OD	PH6OD	PH5OD	PH4OD	PH3OD	PH2OD	PH1OD	PH0OD
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:CMOS 1:open drain							

Port H pull-up control register

PHPUP (0x4000_01EC)

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

Port H pull-down control register

PHPDN (0x4000_01F0)

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

Port H input enable control register

PHIE (0x4000_01F8)

	7	6	5	4	3	2	1	0
Bit Symbol	PH7IE	PH6IE	PH5IE	PH4IE	PH3IE	PH2IE	PH1IE	PH0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0:disabled 1:enabled							

8.2.10 Port I (PI0 to PI1)

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

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

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

Port I register

		7	6	5	4	3	2	1	0
PIDATA	Bit Symbol	–	–	–	–	–	–	PI1	PI0
(0x4000_0200)	Read/Write	R						R/W	
	After reset	"0" is read						"0"	

Port I control register

		7	6	5	4	3	2	1	0
PICR	Bit Symbol	–	–	–	–	–	–	PI1C	PI0C
(0x4000_0204)	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						Output 0: disabled 1: enabled	

Port I open drain control register

		7	6	5	4	3	2	1	0
PIOD	Bit Symbol	–	–	–	–	–	–	PI1OD	PI0OD
(0x4000_ 0228)	Read/Write	R						R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	"0" is read						0:CMOS 1:open drain	

Port I pull-up control register

PIPUP (0x4000_022C)

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

Port I pull-down control register

PIPDN (0x4000_0230)

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

Port I input enable control register

PIIE (0x4000_0238)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	—	PI1IE	PI0IE
Read/Write	R						R/W	
After reset	0	0	0	0	0	0	0	0
Function	"0" is read						Input 0:disabled 1:enabled	

8.2.11 Port J (PJ0 to PJ7)

(Important)
TMPM382 (64 pin version) does not implement port J (PJ0 to PJ7).
Please do not use these functions if you use this product.

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

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

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

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

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

Port J register

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

Port J control register

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

Port J function register 1

		7	6	5	4	3	2	1	0
PJFR1 (0x4000_0248)	Bit Symbol	PJ7F1	PJ6F1	–	–	–	–	–	–
	Read/Write	R/W		R					
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:INTB	0:PORT 1:INTA	“0” is read					

Port J open drain control register

PJOD (0x4000_ 0268)

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

Port J pull-up control register

PJPUP (0x4000_026C)

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

Port J pull-down control register

PJPDN (0x4000_0270)

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

Port J input enable control register

PJIE (0x4000_0278)

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

8.2.12 Port L(PL0, PL2)

The port L is a general-purpose 2-bit port that contained 1-bit output port and 1-bit input/output port. For this port, inputs can be specified in units of bits. Besides the general-purpose input function, the port L performs the functions as the external interrupt input and the operation mode setting.

While a reset signal is in “0”state, the PL0 input and pull-up are enabled. At the rising edge of the reset signal, if PL0 is “1”, the device enters single mode and boots from the on-chip flash memory. If PL0 is “0”, the device enters single boot mode and boots from the internal boot program. For details of single boot mode, refer to “Flash Memory Operation”.

Reset initializes the port L0 as general-purpose output with output disabled , pull-up enabled. Reset initializes the port L2 as general-purpose input/output port with input disabled.

To use the external interrupt input for releasing STOP mode, select this function in the PLFR1 register and enable input in the PLIE register. These settings enable the interrupt input even if the CGSTBYCR<DRVE> bit in the clock/mode control block is set to stop driving of pins during STOP mode.

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

Port L register

		7	6	5	4	3	2	1	0
PLDATA	Bit Symbol	—	—	—	—	—	PL2	–	PL0
(0x4000_02C0)	Read/Write	R					R/W	R	R/W
	After reset	“0” is read					0	“0” is read	0

Port L control register

		7	6	5	4	3	2	1	0
PLCR	Bit Symbol	—	—	—	—	—	PL2C	–	PL0C
(0x4000_02C4)	Read/Write	R					R/W	R	R/W
	After reset	0					0	0	0
	Function	“0” is read.					Output 0: disabled 1: enabled	“0” is read	Output 0: disabled 1: enabled

Port L function register 1

		7	6	5	4	3	2	1	0
PLFR1	Bit Symbol	—	—	—	—	—	PL2F1	–	–
(0x4000_02C8)	Read/Write	R					R/W	R	
	After reset	0					0	0	
	Function	“0” is read					0:PORT 1: INTF	“0” is read	

Port L open drain control register

PLOD (0x4000_ 02E8)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	PL2OD	—	PL0OD
Read/Write	R					R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	“0” is read					0:CMOS 1:open drain	“0” is read	0:CMOS 1:open drain

Port L pull-up control register

PLPUP (0x4000_02EC)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	PL2UP	—	PL0UP
Read/Write	R					R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	“0” is read					Pull-up 0:disabled 1:enabled	0” is read	Pull-up 0:disabled 1:enabled

Port L pull-down control register

PLPDN (0x4000_02F0)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	PL2DN	—	PL0DN
Read/Write	R					R/W		R/W
After reset	0	0	0	0	0	0	0	0
Function	“0” is read					Pull-down 0:disabled 1:enabled	0” is read	Pull-down 0:disabled 1:enabled

Port L input enable control register

PLIE (0x4000_02F8)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	PL2IE	—	—
Read/Write	R					R/W	R	
After reset	0					0	0	0
Function	“0” is read					Input 0:disabled 1:enabled	“0” is read	

8.2.13 Port M (PM0 to PM1)

The port M is a general-purpose, 2-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port I performs the high-speed oscillator1 (X1 and X2).(Note1)

While CGOSCCR<HOSCON> is set to "1",each resister of portM cannot be changed.If you use the external high-speed oscillator,refer to "6.Clock/Mode Control"

Reset initializes all bits of the port M as general-purpose ports with input, output, pull-up, pull-down and high-speed oscillator1 disabled.(Note 2)

(Note 1) The external high-speed oscillator must not be changed while PortM is High output.

(Note 2) After reset,the high-speed clock is selected as the Internal oscillator(OSC2). Therefore the initial state is a general-purpose port M.

Port M register

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

Port M control register

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

Port M open drain control register

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

Port M pull-up control register

PMPUP (0x4000_032C)

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

Port M pull-down control register

PMPDN (0x4000_0330)

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

Port M input enable control register

PMIE (0x4000_0338)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	—	PM1IE	PM0IE
Read/Write	R						R/W	
After reset	0	0	0	0	0	0	0	0
Function	“0” is read						Input 0:disabled 1:enabled	

8.2.14 Port N (PN0 to PN7)

(Important)
TMPM382 (64 pin version) does not implement port N (PN0 to PN7).
Please do not use these functions if you use this product.

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

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

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

Port N register

		7	6	5	4	3	2	1	0
PNDATA	Bit Symbol	PN7	PN6	PN5	PN4	PN3	PN2	PN1	PN0
(0x4000_0340)	Read/Write	R/W							
	After reset	"0"							

Port N control register

		7	6	5	4	3	2	1	0
PNCR	Bit Symbol	PN7C	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
(0x4000_0344)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Output 0: disabled 1: enabled							

Port N functions register 1

		7	6	5	4	3	2	1	0
PNFR1	Bit Symbol	PN7F1	PN6F1	PN5F1	PN4F1	PN3F1	PN2F1	PN1F1	PN0F1
(0x4000_0348)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:MT2IN	0:PORT 1:$\overline{\text{EMG2}}$	0:PORT 1:MTOUT12	0:PORT 1: MTOUT02	0:PORT 1:SP1FSS	0:PORT 1:SP1CLK	0:PORT 1:SP1DI	0:PORT 1: SP1DO0

Port N functions register 2

PNFR2 (0x4000_034C)

	7	6	5	4	3	2	1	0
Bit Symbol	PN7F2	−	PN5F2	PN4F2	−	−	−	−
Read/Write	R/W	R	R/W	R/W	R			
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:INTE.	“0” is read.	0:PORT 1:MTTB2IN	0:PORT 1:MTTB2OUT	“0” is read.			

Port N open drain control register

PNOD (0x4000_ 0368)

	7	6	5	4	3	2	1	0
Bit Symbol	PN7OD	PN6OD	PN5OD	PN4OD	PN3OD	PN2OD	PN1OD	PN0OD
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:CMOS 1:Open drain							

Port N pull-up control register

PNPUP (0x4000_036C)

	7	6	5	4	3	2	1	0
Bit Symbol	PN7UP	PN6UP	PN5UP	PN4UP	PN3UP	PN2UP	PN1UP	PN0UP
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0:disabled 1:enabled							

Port N pull-down control register

PNPDN (0x4000_0370)

	7	6	5	4	3	2	1	0
Bit Symbol	PN7DN	PN6DN	PN5DN	PN4DN	PN3DN	PN2DN	PN1DN	PN0DN
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-down 0:disabled 1:enabled							

Port N input enable control register

PNIE (0x4000_0378)

	7	6	5	4	3	2	1	0
Bit Symbol	PN7IE	PN6IE	PN5IE	PN4IE	PN3IE	PN2IE	PN1IE	PN0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0:disabled 1:enabled							

8.2.15 Port P (PP0 to PP1)

The port P is a general-purpose, 2-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Besides the general-purpose input/output function, the port P performs the low-speed oscillator(XT1 and XT2). (Note 1)

Reset initializes all bits of the port P as general-purpose ports with input, output, pull-up, pull-down and low-speed oscillator disabled.(Note 2)

(Note 1) The external low-speed oscillator must not be changed while PortP is High output.
If you use the external low-speed oscillator,refer to "6.Clock/Mode Control"

(Note 2) After reset,the low-speed clock is stopped.
Therefore the initial state is a general-purpose port P.

Port P register

PPDATA (0x4000_0380)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	–	PP1	PP0
Read/Write	R						R/W	
After reset	“0” is read						“0”	

Port P control register

PPCR (0x4000_0384)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	–	PP1C	PP0C
Read/Write	R						R/W	
After reset	0						0	0
Function	“0” is read.						Output 0: disabled 1: enabled	

Port P open drain control register

PPOD (0x4000_ 03A8)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	–	—	PP1OD	PP0OD
Read/Write	R						R/W	
After reset	0	0	0	0	0	0	0	0
Function	“0” is read						0:CMOS 1:open drain	

Port P pull-up control register

PPPUP (0x4000_03AC)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	—	PP1UP	PP0UP
Read/Write	R						R/W	
After reset	0	0	0	0	0	0	0	0
Function	“0” is read						Pull-up 0:disabled 1:enabled	

Port P pull-down control register

PPPDN (0x4000_03B0)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	—	PP1DN	PP0DN
Read/Write	R						R/W	
After reset	0	0	0	0	0	0	0	0
Function	“0” is read						Pull-down 0:disabled 1:enabled	

Port P input enable control register

PPIE (0x4000_03B8)

	7	6	5	4	3	2	1	0
Bit Symbol	—	—	—	—	—	—	PP1IE	PP0IE
Read/Write	R						R/W	
After reset	0						0	0
Function	“0” is read						Input 0:disabled 1:enabled	

8.3 Appendix (Ports setting)

The setting of the registers of each function lists are shown in the following.

8.3.1 Port A setting

Table 8-2 Port A setting

Port Name	Function	Type	Port A SFR						
			PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE
PA0	* GPIO	-	★	★	0	★	★	★	★
	TB0IN	FT1	X	0	PA0F1	0	★	★	1
	INT3	FT4	X	0	PA0F2	0	★	★	1
PA1	* GPIO	-	★	★	0	★	★	★	★
	TB0OUT	FT1	X	1	PA1F1	★	★	★	0
	SCOUT	FT1	X	1	PA1F2	★	★	★	0
PA2	* GPIO	-	★	★	0	★	★	★	★
	TB1IN	FT1	X	0	PA2F1	0	★	★	1
	INT4	FT4	X	0	PA2F2	0	★	★	1
PA3	* GPIO	-	★	★	0	★	★	★	★
	TB1OUT	FT1	X	1	PA3F1	★	★	★	0
	RXIN	FT1	X	0	PA3F2	0	★	★	1
PA4	* GPIO	-	★	★	0	★	★	★	★
	SCLK1 (IN)	FT1	X	0	PA4F1	0	★	★	1
	SCLK1 (OUT)	FT1	X	1	PA4F1	★	★	★	0
	CTS1	FT1	X	0	PA4F2	0	★	★	1
PA5	* GPIO	-	★	★	0	★	★	★	★
	TX1	FT1	X	1	PA5F1	★	★	★	0
	TB6OUT	FT1	X	1	PA5F2	★	★	★	0
PA6	* GPIO	-	★	★	0	★	★	★	★
	RX1	FT1	X	0	PA6F1	0	★	★	1
	TB6IN	FT1	X	0	PA6F2	0	★	★	1
PA7	* GPIO	-	★	★	0	★	★	★	★
	TB4IN	FT1	X	0	PA7F1	0	★	★	1
	INT8	FT4	X	0	PA7F2	0	★	★	1

【Function】

*: Initial state

INTn: When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PAFR2 register.

【PAFRn】

"0": All the corresponding bits of the PAFRn registers are not selected.

"PAxFn": The bits of the PAFRn registers should be selected are described by the bit symbol name.

【 Common setting in Port A SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.2 Port B setting

Table 8-3 Port B setting

Port Name	Function	Type	Port B SFR						
			PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE
PB0	* GPIO	-	★	★	0	★	★	★	★
	TRACECLK	FT1	X	1	PB0F1	★	★	★	0
PB1	* GPIO	-	★	★	0	★	★	★	★
	TRACEDATA0	FT1	X	1	PB1F1	★	★	★	0
PB2	* GPIO	-	★	★	0	★	★	★	★
	TRACEDATA1	FT1	X	1	PB2F1	★	★	★	0
PB3	* TMS/SWDIO	FT2	X	1	PB3F1	0	1	0	1
	GPIO	-	★	★	0	★	★	★	★
PB4	* TCK/SWCLK	FT2	X	0	PB4F1	0	0	1	1
	GPIO	-	★	★	0	★	★	★	★
PB5	* TDO/SWV	FT2	X	1	PB5F1	0	0	0	0
	GPIO	-	★	★	0	★	★	★	★
PB6	* TDI	FT2	X	0	PB6F1	0	1	0	1
	GPIO	-	★	★	0	★	★	★	★
PB7	* $\overline{\text{TRST}}$	FT2	X	0	PB7F1	0	1	0	1
	GPIO	-	★	★	0	★	★	★	★

【Function】

*: Initial state

【PBFRn】

"0": All the corresponding bits of the PBFR1 registers are not selected.

"PBxFn": The bits of the PBFR1 registers should be selected are described by the bit symbol name.

【 Common setting in Port C SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.3 Port C setting

Table 8-4 Port C setting

Port Name	Function	Type	Port C SFR						
			PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE
PC0	* GPIO	-	★	★	0	★	★	★	★
	UO0	FT3	X	1	PC0F1	★	★	★	0
	SP0DO	FT3	X	1	PC0F2	★	★	★	0
	SDA0	FT1	X	1	PC0F3	1	★	★	1
	SO0	FT1	X	1	PC0F4	★	★	★	0
PC1	* GPIO	-	★	★	0	★	★	★	★
	XO0	FT3	X	1	PC1F1	★	★	★	0
	SP0DI	FT1	X	0	PC1F2	0	★	★	1
	SCL0	FT1	X	1	PC1F3	1	★	★	1
	SI0	FT1	X	0	PC1F4	0	★	★	1
PC2	* GPIO	-	★	★	0	★	★	★	★
	VO0	FT3	X	1	PC2F1	★	★	★	0
	SP0CLK (IN)	FT3	X	0	PC2F2	0	★	★	1
	SP0CLK (OUT)	FT3	X	1	PC2F2	★	★	★	0
	SCK0 (IN)	FT1	X	0	PC2F3	0	★	★	1
	SCK0 (OUT)	FT1	X	1	PC2F3	★	★	★	0
PC3	* GPIO	-	★	★	0	★	★	★	★
	YO0	FT3	X	1	PC3F1	★	★	★	0
	SP0FSS (IN)	FT3	X	0	PC3F2	0	★	★	1
	SP0FSS (OUT)	FT3	X	1	PC3F2	★	★	★	0
PC4	* GPIO	-	★	★	0	★	★	★	★
	WO0	FT3	X	1	PC4F1	★	★	★	0
	MTOUT00	FT3	X	1	PC4F2	★	★	★	0
	MTTB0OUT	FT1	X	1	PC4F3	★	★	★	0
PC5	* GPIO	-	★	★	0	★	★	★	★
	ZO0	FT3	X	1	PC5F1	★	★	★	0
	MTOUT10	FT3	X	1	PC5F2	★	★	★	0
	MTTB0IN	FT1	X	0	PC5F3	0	★	★	1
	SCLK4 (IN)	FT1	X	0	PC5F4	0	★	★	1
	SCLK4 (OUT)	FT1	X	1	PC5F4	★	★	★	0
	CTS4	FT1	X	0	PC5F5	0	★	★	1
PC6	* GPIO	-	★	★	0	★	★	★	★
	EMG0	FT1	X	0	PC6F1	0	★	★	1
	GEMG0	FT1	X	0	PC6F2	0	★	★	1
	TX4	FT1	X	1	PC6F4	★	★	★	0
PC7	* GPIO	-	★	★	0	★	★	★	★
	MT0IN	FT1	X	0	PC7F2	0	★	★	1
	RX4	FT1	X	0	PC7F4	0	★	★	1

【Function】

*: Initial state

【PCFRn】

"0": All the corresponding bits of the PCFRn registers are not selected.

"PCxFn": The bits of the PCFRn registers should be selected are described by the bit symbol name.

【 Common setting in Port C SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.4 Port D setting

Table 8-5 Port D setting

Port Name	Function	Type	Port D SFR						
			PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE
PD0	* GPIO	-	★	★	0	★	★	★	★
	ENCA0	FT1	X	0	PD0F1	0	★	★	1
	TB5IN	FT1	X	0	PD0F2	0	★	★	1
	INTC	FT4	X	0	PD0F3	0	★	★	1
PD1	* GPIO	-	★	★	0	★	★	★	★
	ENCB0	FT1	X	0	PD1F1	0	★	★	1
	TB5OUT	FT1	X	1	PD1F2	★	★	★	0
PD2	* GPIO	-	★	★	0	★	★	★	★
	ENCZ0	FT1	X	0	PD2F1	0	★	★	1
	INTD	FT4	X	0	PD2F3	0	★	★	1
PD3	* GPIO	-	★	★	0	★	★	★	★
	INT9	FT4	X	0	PD3F1	0	★	★	1
PD4	* GPIO	-	★	★	0	★	★	★	★
	SCLK2 (IN)	FT1	X	0	PD4F1	0	★	★	1
	SCLK2 (OUT)	FT1	X	1	PD4F1	★	★	★	0
	CTS2	FT1	X	0	PD4F2	0	★	★	1
PD5	* GPIO	-	★	★	0	★	★	★	★
	TX2	FT1	X	1	PD5F1	★	★	★	0
PD6	* GPIO	-	★	★	0	★	★	★	★
	RX2	FT1	X	0	PD6F1	0	★	★	1

【Function】

*: Initial state

INTn: When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PDFRn register.

【PDFRn】

"0": All the corresponding bits of the PDFRn registers are not selected.

"PDxFn": The bits of the PDFRn registers should be selected are described by the bit symbol name.

【 Common setting in Port D SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.5 Port E setting

Table 8-6 Port E setting

Port Name	Function	Type	Port E SFR						
			PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0	* GPIO	-	★	★	0	★	★	★	★
	TX0	FT1	X	1	PE0F1	★	★	★	0
PE1	* GPIO	-	★	★	0	★	★	★	★
	RX0	FT1	X	0	PE1F1	0	★	★	1
PE2	* GPIO	-	★	★	0	★	★	★	★
	SCLK0 (IN)	FT1	X	0	PE2F1	0	★	★	1
	SCLK0 (OUT)	FT1	X	1	PE2F1	★	★	★	0
	CTS0	FT1	X	0	PE2F2	0	★	★	1
PE3	* GPIO	-	★	★	0	★	★	★	★
	TB4OUT	FT1	X	1	PE3F1	★	★	★	0
PE4	* GPIO	-	★	★	0	★	★	★	★
	TB2IN	FT1	X	0	PE4F1	0	★	★	1
	INT5	FT4	X	0	PE4F2	0	★	★	1
PE5	* GPIO	-	★	★	0	★	★	★	★
	TB2OUT	FT1	X	1	PE5F1	★	★	★	0
PE6	* GPIO	-	★	★	0	★	★	★	★
	TB3IN	FT1	X	0	PE6F1	0	★	★	1
	INT6	FT4	X	0	PE6F2	0	★	★	1
PE7	* GPIO	-	★	★	0	★	★	★	★
	TB3OUT	FT1	X	1	PE7F1	★	★	★	0
	INT7	FT4	X	0	PE7F2	0	★	★	1

【Function】

*: Initial state

INTn: When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PEFR2 register.

【PEFRn】

"0": All the corresponding bits of the PEFRn registers are not selected.

"PExFn": The bits of the PEFRn registers should be selected are described by the bit symbol name.

【 Common setting in Port E SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.6 Port F setting

Table 8-7 Port F setting

Port Name	Function	Type	Port F SFR						
			PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE
PF0	* GPIO	-	★	★	0	★	★	★	★
	TB7IN	FT1	X	0	PF0F1	0	★	★	1
PF1	* GPIO	-	★	★	0	★	★	★	★
	TB7OUT	FT1	X	1	PF1F1	★	★	★	0
	$\overline{\text{ALARM}}$	FT1	X	1	PF1F2	★	★	★	0
PF2	* GPIO	-	★	★	0	★	★	★	★
	ENCA1	FT1	X	0	PF2F1	0	★	★	1
	SCLK3 (IN)	FT1	X	0	PF2F2	0	★	★	1
	SCLK3 (OUT)	FT1	X	1	PF2F2	★	★	★	0
	CTS3	FT1	X	0	PF2F3	0	★	★	1
PF3	* GPIO	-	★	★	0	★	★	★	★
	ENCB1	FT1	X	0	PF3F1	0	★	★	1
	TX3	FT1	X	1	PF3F2	★	★	★	0
PF4	* GPIO	-	★	★	0	★	★	★	★
	ENCZ1	FT1	X	0	PF4F1	0	★	★	1
	RX3	FT1	X	0	PF4F2	0	★	★	1

【Function】

*: Initial state

【PFFRn】

"0": All the corresponding bits of the PFFRn registers are not selected.

"PFxFn": The bits of the PFFRn registers should be selected are described by the bit symbol name.

【 Common setting in Port F SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.7 Port G setting

Table 8-8 Port G setting

Port Name	Function	Type	Port G SFR						
			PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE
PG0	* GPIO	-	★	★	0	★	★	★	★
	UO1	FT3	X	1	PG0F1	★	★	★	0
	SDA1	FT1	X	1	PG0F3	1	★	★	1
	SO1	FT1	X	1	PG0F4	★	★	★	0
PG1	* GPIO	-	★	★	0	★	★	★	★
	XO1	FT3	X	1	PG1F1	★	★	★	0
	SCL1	FT1	X	1	PG1F3	1	★	★	1
	SI1	FT1	X	0	PG1F4	0	★	★	1
PG2	* GPIO	-	★	★	0	★	★	★	★
	VO1	FT3	X	1	PG2F1	★	★	★	0
	SCK1 (IN)	FT1	X	0	PG2F3	0	★	★	1
	SCK1 (OUT)	FT1	X	1	PG2F4	★	★	★	0
PG3	* GPIO	-	★	★	0	★	★	★	★
	YO1	FT3	X	1	PG3F1	★	★	★	0
PG4	* GPIO	-	★	★	0	★	★	★	★
	WO1	FT3	X	1	PG4F1	★	★	★	0
	MTOUT01	FT3	X	1	PG4F2	★	★	★	0
	MTTB1OUT	FT1	X	1	PG4F3	★	★	★	0
PG5	* GPIO	-	★	★	0	★	★	★	★
	ZO1	FT3	X	1	PG5F1	★	★	★	0
	MTOUT11	FT3	X	1	PG5F2	★	★	★	0
	MTTB1IN	FT1	X	0	PG5F3	0	★	★	1
PG6	* GPIO	-	★	★	0	★	★	★	★
	$\overline{\text{EMG1}}$	FT1	X	0	PG6F1	0	★	★	1
	$\overline{\text{GEMG1}}$	FT1	X	0	PG6F2	0	★	★	1
PG7	* GPIO	-	★	★	0	★	★	★	★
	MT1IN	FT1	X	0	PG7F2	0	★	★	1

【Function】

*: Initial state

【PGFRn】

"0": All the corresponding bits of the PGFRn registers are not selected.

"PGxFn": The bits of the PGFRn registers should be selected are described by the bit symbol name.

【 Common setting in Port G SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.8 Port H setting

Table 8-9 Port H setting

Port Name	Function	Type	Port H SFR						
			PHDATA	PHCR	PHFRn	PHOD	PHPUP	PHPDN	PHIE
PH0	* GPIO	-	★	★	0	★	★	★	★
	INT0	FT4	X	0	PH0F1	0	★	★	1
	AIN0	FT5	X	0	0	0	0	0	0
PH1	* GPIO	-	★	★	0	★	★	★	★
	INT1	FT4	X	0	PH1F1	0	★	★	1
	AIN1	FT5	X	0	0	0	0	0	0
PH2	* GPIO	-	★	★	0	★	★	★	★
	INT2	FT4	X	0	PH2F1	0	★	★	1
	AIN2	FT5	X	0	0	0	0	0	0
PH3	* GPIO	-	★	★	-	★	★	★	★
	AIN3	FT5	X	0	-	0	0	0	0
PH4	* GPIO	-	★	★	-	★	★	★	★
	AIN4	FT5	X	0	-	0	0	0	0
PH5	* GPIO	-	★	★	-	★	★	★	★
	AIN5	FT5	X	0	-	0	0	0	0
PH6	* GPIO	-	★	★	-	★	★	★	★
	AIN6	FT5	X	0	-	0	0	0	0
PH7	* GPIO	-	★	★	-	★	★	★	★
	AIN7	FT5	X	0	-	0	0	0	0

【Function】

*: Initial state

INTn: When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PHFR1 register.

【PHFR1】

"0": All the corresponding bits of the PHFR1 registers are not selected.

"PHxF1": The bits of the PHFR1 registers should be selected are described by the bit symbol name.

-: There is no corresponding bit.

【Common setting in Port H SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.9 Port I setting

Table 8-10 Port I setting

Port Name	Function	Type	Port I SFR						
			PIDATA	PICR	PIFRn	PIOD	PIPUP	PIPDN	PIIE
PI0	* GPIO	-	★	★	-	★	★	★	★
	AIN8	FT5	X	0	-	0	0	0	0
PI1	* GPIO	-	★	★	-	★	★	★	★
	AIN9	FT5	X	0	-	0	0	0	0

【Function】

*: Initial state

【PIFR1】

"0": All the corresponding bits of the PIFR1 registers are not selected.

"PIxF1": The bits of the PIFR1 registers should be selected are described by the bit symbol name.

-: There is no corresponding bit.

【Common setting in Port I SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

-: There is no corresponding bit.

8.3.10 Port J setting

Table 8-11 Port J setting

Port Name	Function	Type	Port J SFR						
			PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE
PJ0	* GPIO	-	★	★	-	★	★	★	★
	AIN10	FT5	X	0	-	0	0	0	0
PJ1	* GPIO	-	★	★	-	★	★	★	★
	AIN11	FT5	X	0	-	0	0	0	0
PJ2	* GPIO	-	★	★	-	★	★	★	★
	AIN12	FT5	X	0	-	0	0	0	0
PJ3	* GPIO	-	★	★	-	★	★	★	★
	AIN13	FT5	X	0	-	0	0	0	0
PJ4	* GPIO	-	★	★	-	★	★	★	★
	AIN14	FT5	X	0	-	0	0	0	0
PJ5	* GPIO	-	★	★	-	★	★	★	★
	AIN15	FT5	X	0	-	0	0	0	0
PJ6	* GPIO	-	★	★	0	★	★	★	★
	INTA	FT4	X	0	PJ6F1	0	★	★	1
	AIN16	FT5	X	0	0	0	0	0	0
PJ7	* GPIO	-	★	★	0	★	★	★	★
	INTB	FT4	X	0	PJ7F1	0	★	★	1
	AIN17	FT5	X	0	0	0	0	0	0

【Function】

*: Initial state

INTn: When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PJFR1 register.

【PJFR1】

"0": All the corresponding bits of the PJFR1 registers are not selected.

"PJxF1": The bits of the PJFR1 registers should be selected are described by the bit symbol name.

-: There is no corresponding bit.

【Common setting in Port J SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.11 Port L setting

Table 8-12 Port L setting

Port Name	Function	Type	Port L SFR						
			PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE
PL0	* GPIO	-	★	★	-	★	★	★	★
	BOOT	FT6	X	X	-	X	X	X	X
PL2	* GPIO	-	★	★	0	★	★	★	★
	INTF	FT4	X	0	PL2F1	0	★	★	1

【Function】

*: Initial state

●: The function that becomes effective for reset.

INTF: When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PLFR1 register.

【PLFR1】

"0": All the corresponding bits of the PLFR1 registers are not selected.

"PL2F1": The bits of the PLFR1 registers should be selected are described by the bit symbol name.

-: There is no corresponding bit.

【Common setting in Port L SFR】

"0": The corresponding bit is set to “0”.

"1": The corresponding bit is set to “1”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.3.12 Port M setting

Table 8-13 Port M setting

Port Name	Function	Type	Port M SFR						
			PMDATA	PMCR	PMFRn	PMOD	PMPUP	PMPDN	PMIE
PM0	* GPIO	-	★	★	-	★	★	★	★
	X1	FT5	X	0	-	0	0	0	0
PM1	* GPIO	-	★	★	-	★	★	★	★
	X2	FT5	X	0	-	0	0	0	0

【Function】

*：Initial state

【PMFR1】

-：There is no corresponding bit.

【Common setting in Port M SFR】

"0"：The corresponding bit is set to “0”.

★:Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X：Don't Care (After reset "0")

8.3.13 Port N setting

Table 8-14 Port N setting

Port Name	Function	Type	Port N SFR						
			PNDATA	PNCR	PNFRn	PNOD	PNPUP	PNPDN	PNIE
PN0	* GPIO	-	★	★	0	★	★	★	★
	SP1DO	FT3	X	1	PN0F1	★	★	★	0
PN1	* GPIO	-	★	★	0	★	★	★	★
	SP1DI	FT1	X	0	PN1F1	0	★	★	1
PN2	* GPIO	-	★	★	0	★	★	★	★
	SP1CLK (IN)	FT3	X	0	PN2F1	0	★	★	1
	SP1CLK (OUT)	FT3	X	1	PN2F1	★	★	★	0
PN3	* GPIO	-	★	★	0	★	★	★	★
	SP1FSS (IN)	FT3	X	0	PN3F1	0	★	★	1
	SP1FSS (OUT)	FT3	X	1	PN3F1	★	★	★	0
PN4	* GPIO	-	★	★	0	★	★	★	★
	MTOUT02	FT3	X	1	PN4F1	★	★	★	0
	MTTB2OUT	FT1	X	1	PN4F2	★	★	★	0
PN5	* GPIO	-	★	★	0	★	★	★	★
	MTOUT12	FT3	X	1	PN5F1	★	★	★	0
	MTTB2IN	FT1	X	0	PN5F2	0	★	★	1
PN6	* GPIO	-	★	★	0	★	★	★	★
	$\overline{\text{GEMG2}}$	FT1	X	0	PN6F1	0	★	★	1
PN7	* GPIO	-	★	★	0	★	★	★	★
	MT2IN	FT1	X	0	PN7F1	0	★	★	1
	INTE	FT4	X	0	PN7F2	0	★	★	1

【Function】

* : Initial state

INTE : When the external interrupt input for releasing STOP mode is not used, It is not necessary to care for the PNFR2 register.

【PNFRn】

"0" : All the corresponding bits of the PNFRn registers are not selected.

"PNxFn" : The bits of the PNFRn registers should be selected are described by the bit symbol name.

【Common setting in Port N SFR】

"0" : The corresponding bit is set to “0”.

"1" : The corresponding bit is set to “1”.

□ : Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X : Don't Care (After reset "0")

8.3.14 Port P setting

Table 8-15 Port P setting

Port Name	Function	Type	Port P SFR						
			PPDATA	PPCR	PPFRn	PPOD	PPPUP	PPPDN	PPIE
PP0	* GPIO	-	★	★	-	★	★	★	★
	XT1	FT5	X	0	-	0	0	0	0
PP1	* GPIO	-	★	★	-	★	★	★	★
	XT2	FT5	X	0	-	0	0	0	0

【Function】

*: Initial state

【PPFRn】

-: There is no corresponding bit.

【Port P SFR 共通】

"0": The corresponding bit is set to “0”.

★: Arbitrarily it sets and it uses it according to the purpose. (After reset "0")

X: Don't Care (After reset "0")

8.4 Port Section Equivalent Circuit Schematics

The setting of the registers of each function list is shown in the following.

8.4.1 Port type FT1

<I/O> SCLKn/SCKn/SCLn/SDAn

<In> TBnIN/RXn/CTSn/Sin/SPnDI/RXIN/EMGn/GEMGn/MTnIN/MTTBnIN/ENCAn/ ENCBn/ ENCZn

<Out> TBnOUT/TXn/Son/SCOUT/ALARM/MTTBnOUT/TRACECLK/TRACEDATAn

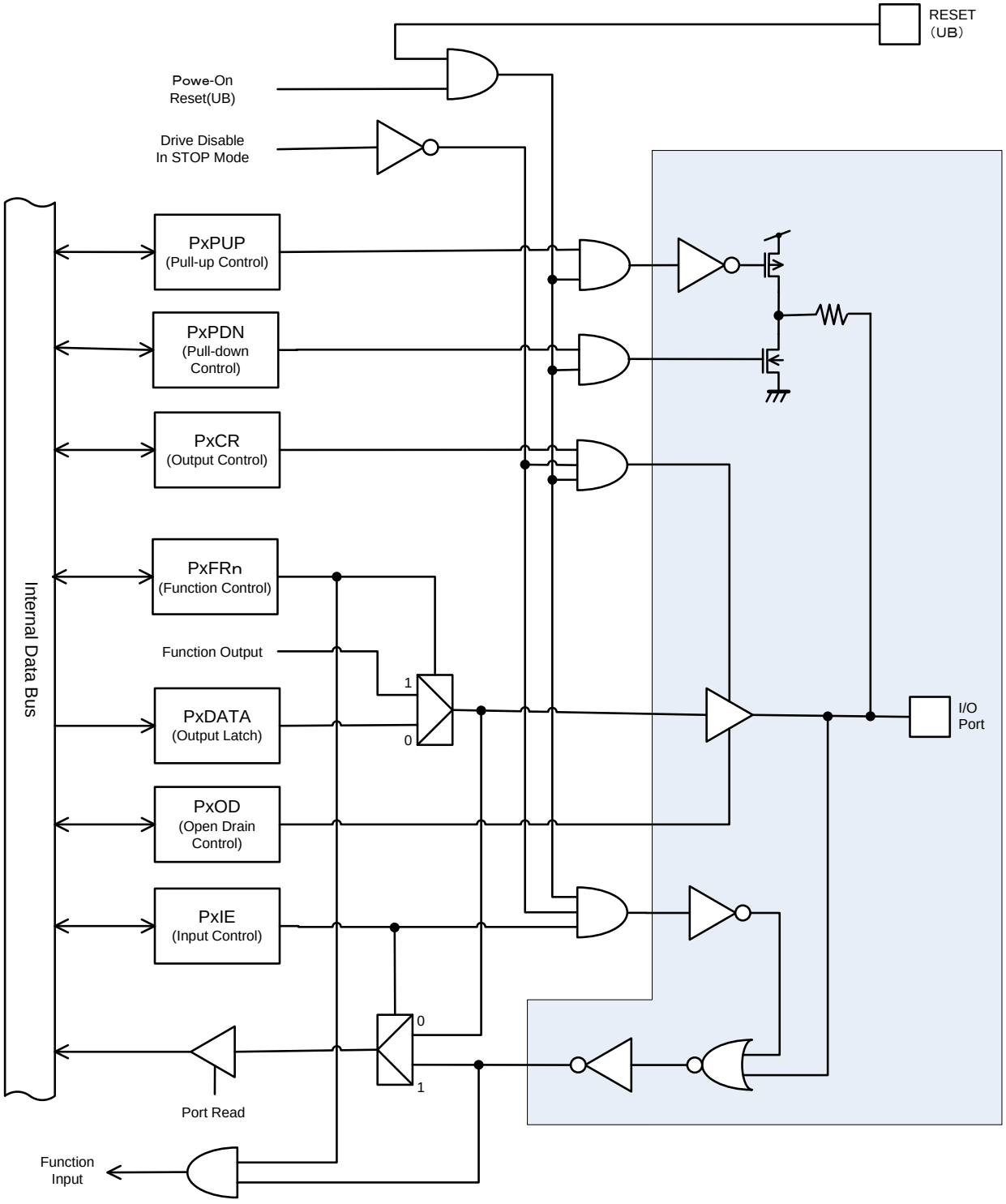


Figure 8-1 Port Type FT1

8.4.2 Port type FT2

<I/O> TMS/SWDIO
<In> TCK/SWCLK/TDI/TRST
<Out> TDO/SWV+Pio

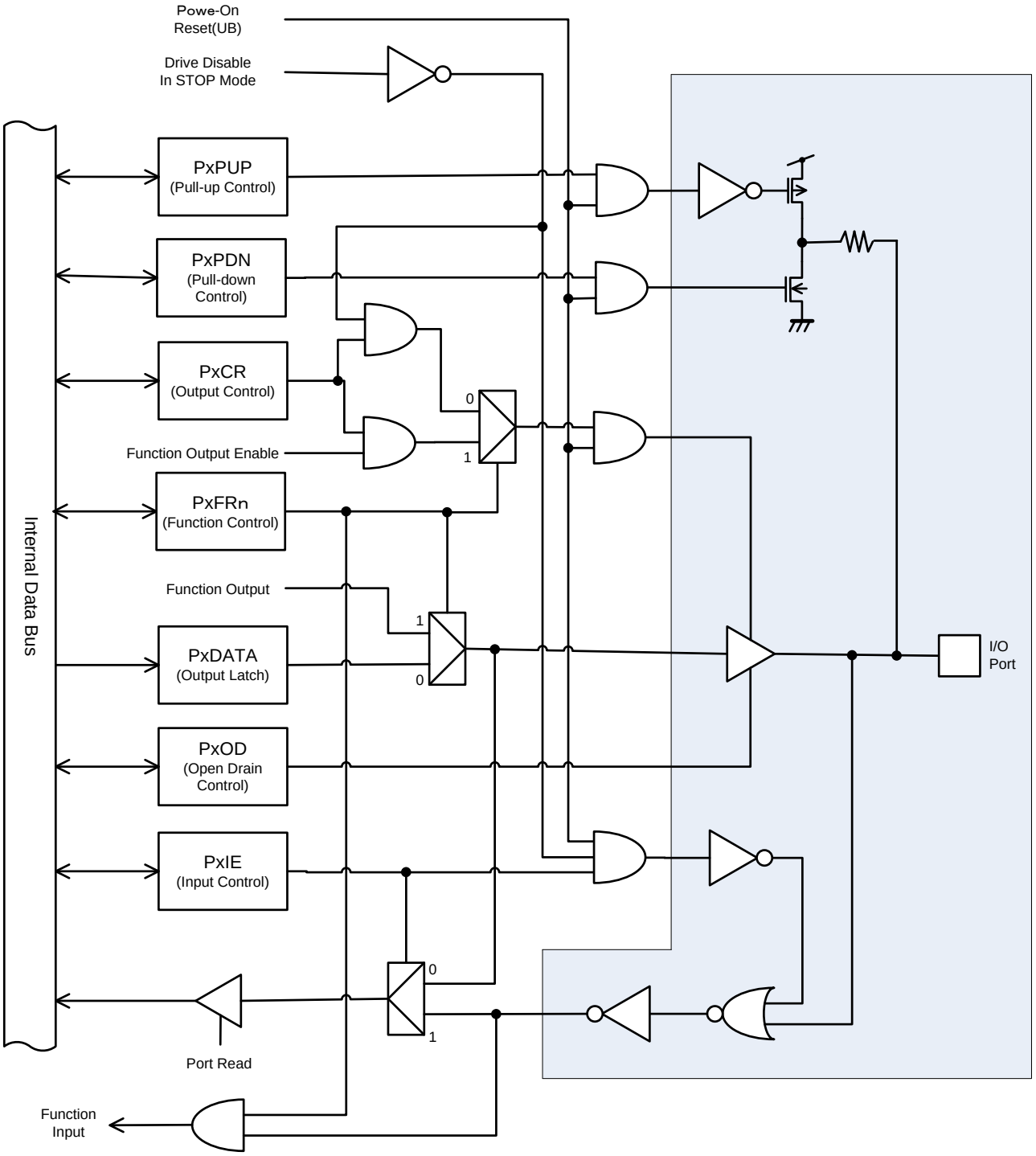


Figure 8-2 Port Type FT2

8.4.3 Port type FT3

<I/O> SPnCLK/SPnFSS
<Out> SPnDO/UOn/VOn/WOn/YOn/ZOn/MTOUT0n/MTOUT1n

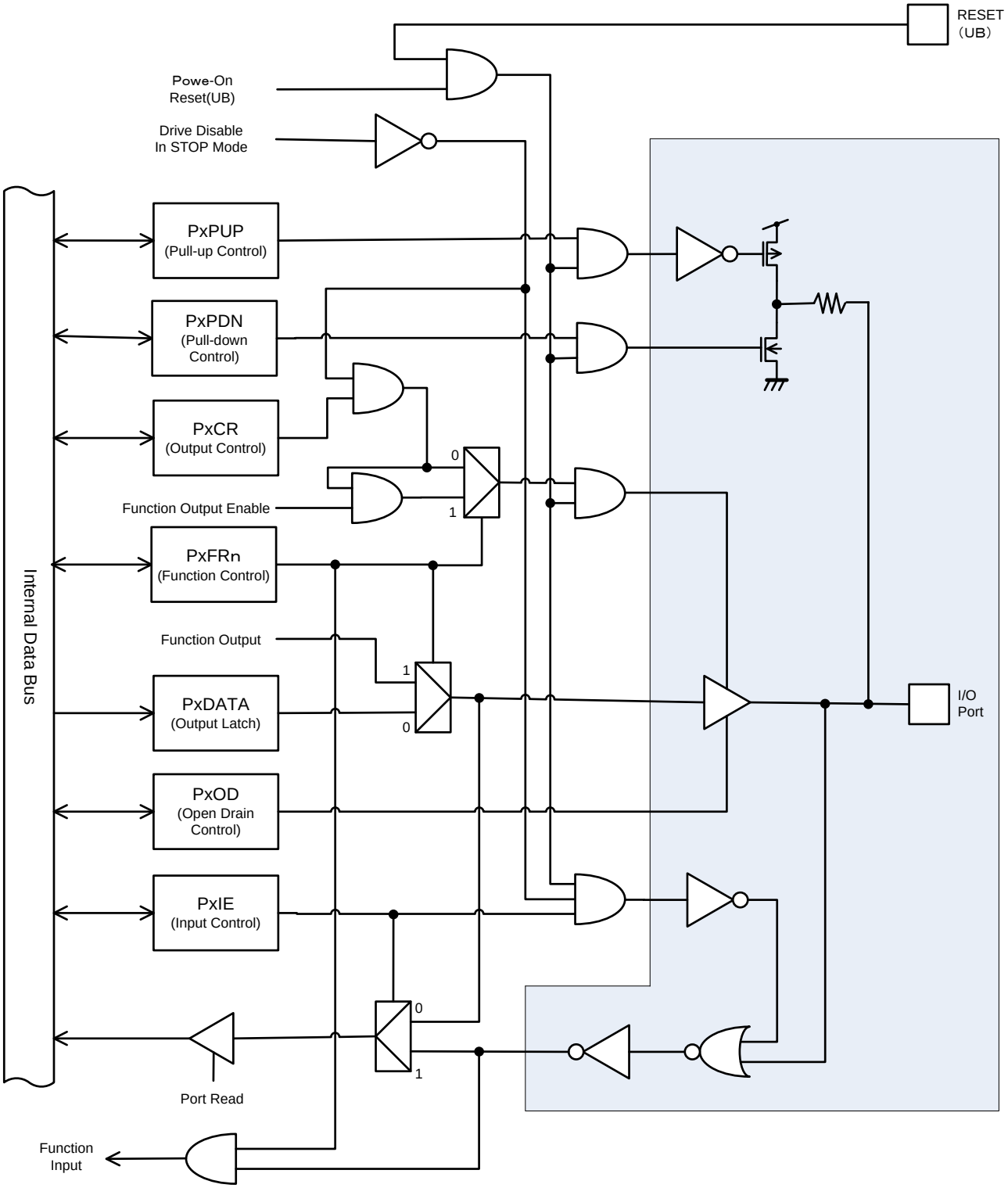


Figure 8-3 Port Type FT3

8.4.4 Port type FT4

<In> INTn

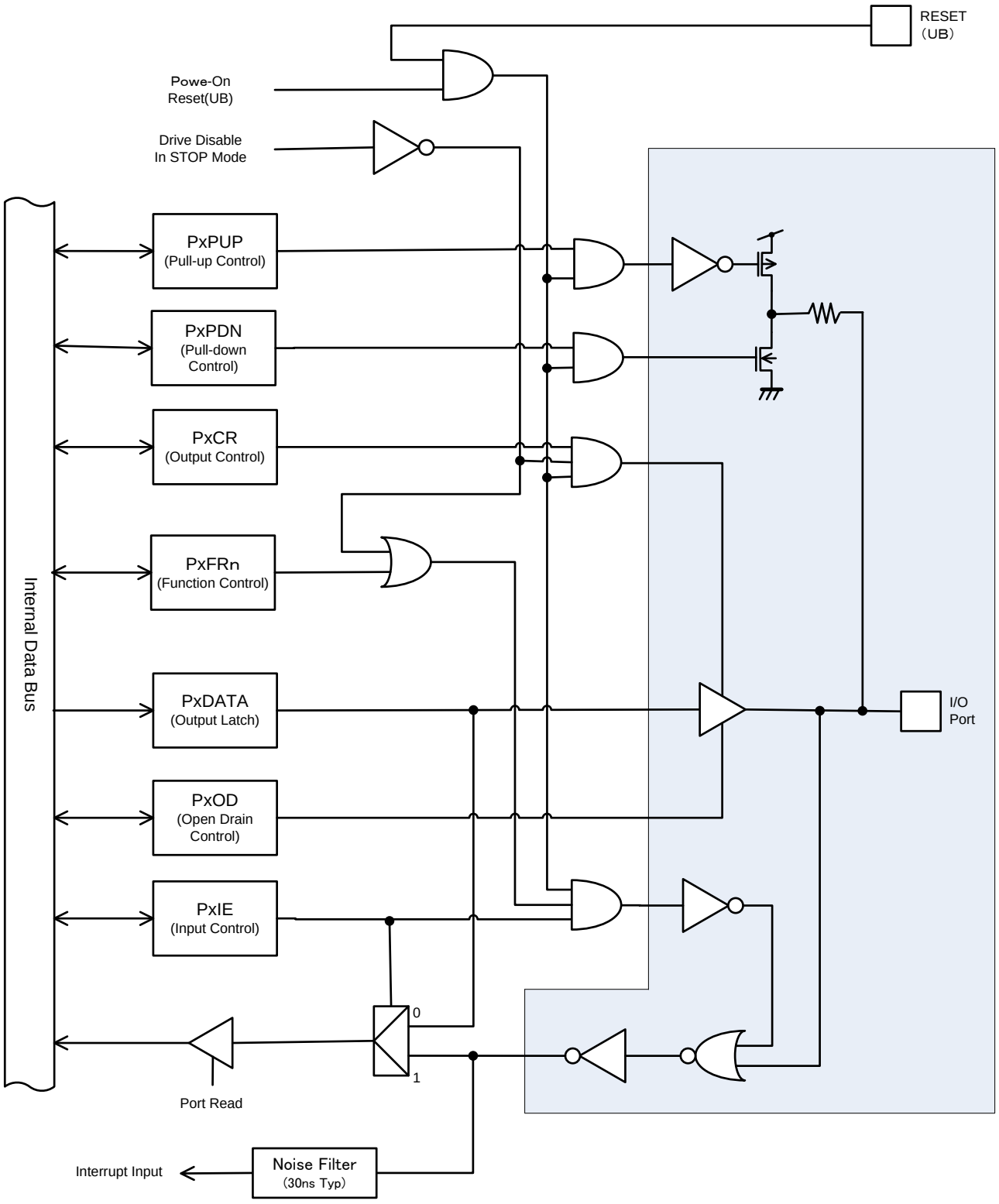


Figure 8-4 Port Type FT4

8.4.5 Port type FT5

<In> AINn/Xn/XTn

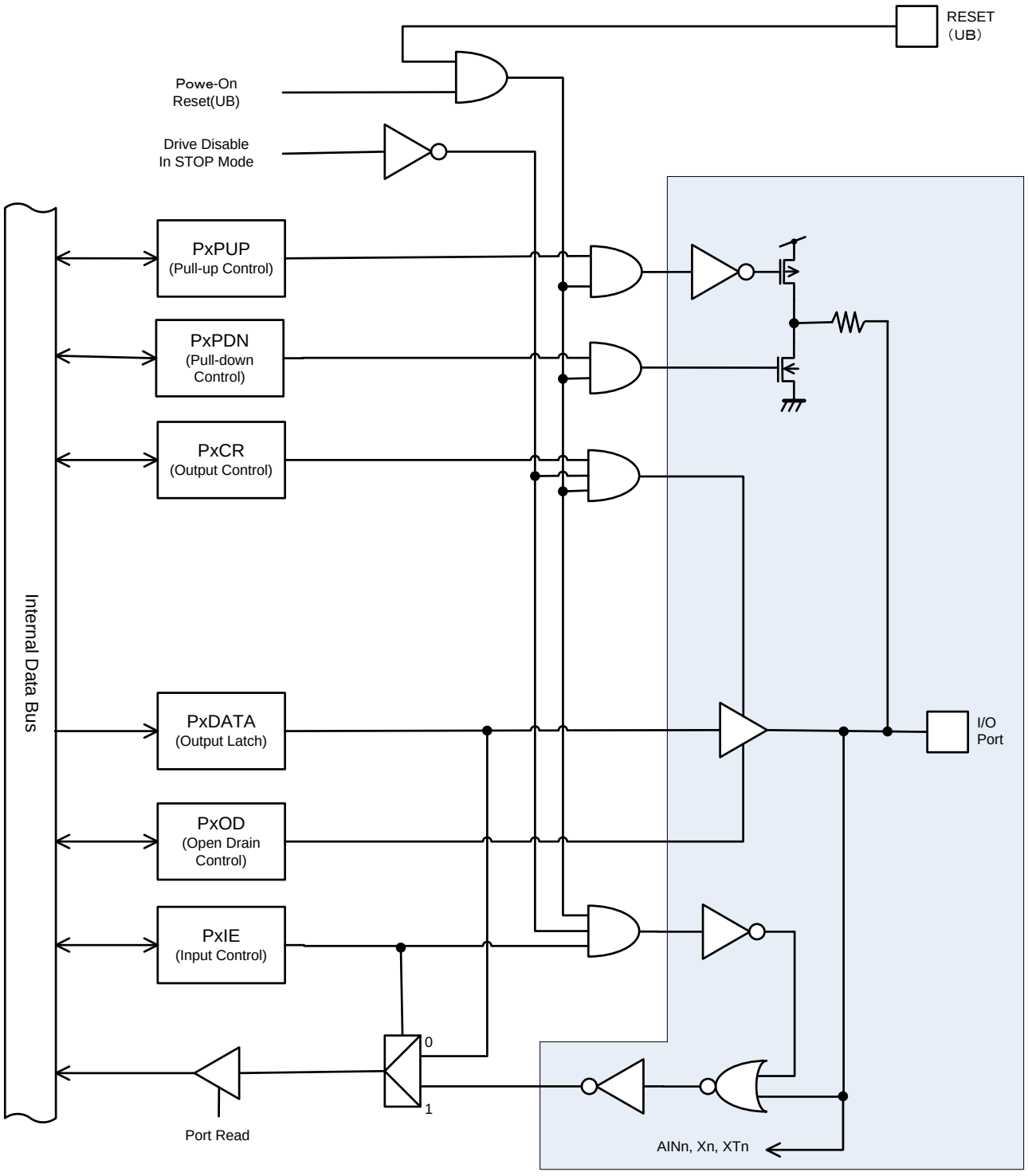


Figure 8-5 Port Type FT5

8.4.6 Port type FT6

<In> $\overline{\text{BOOT}}$

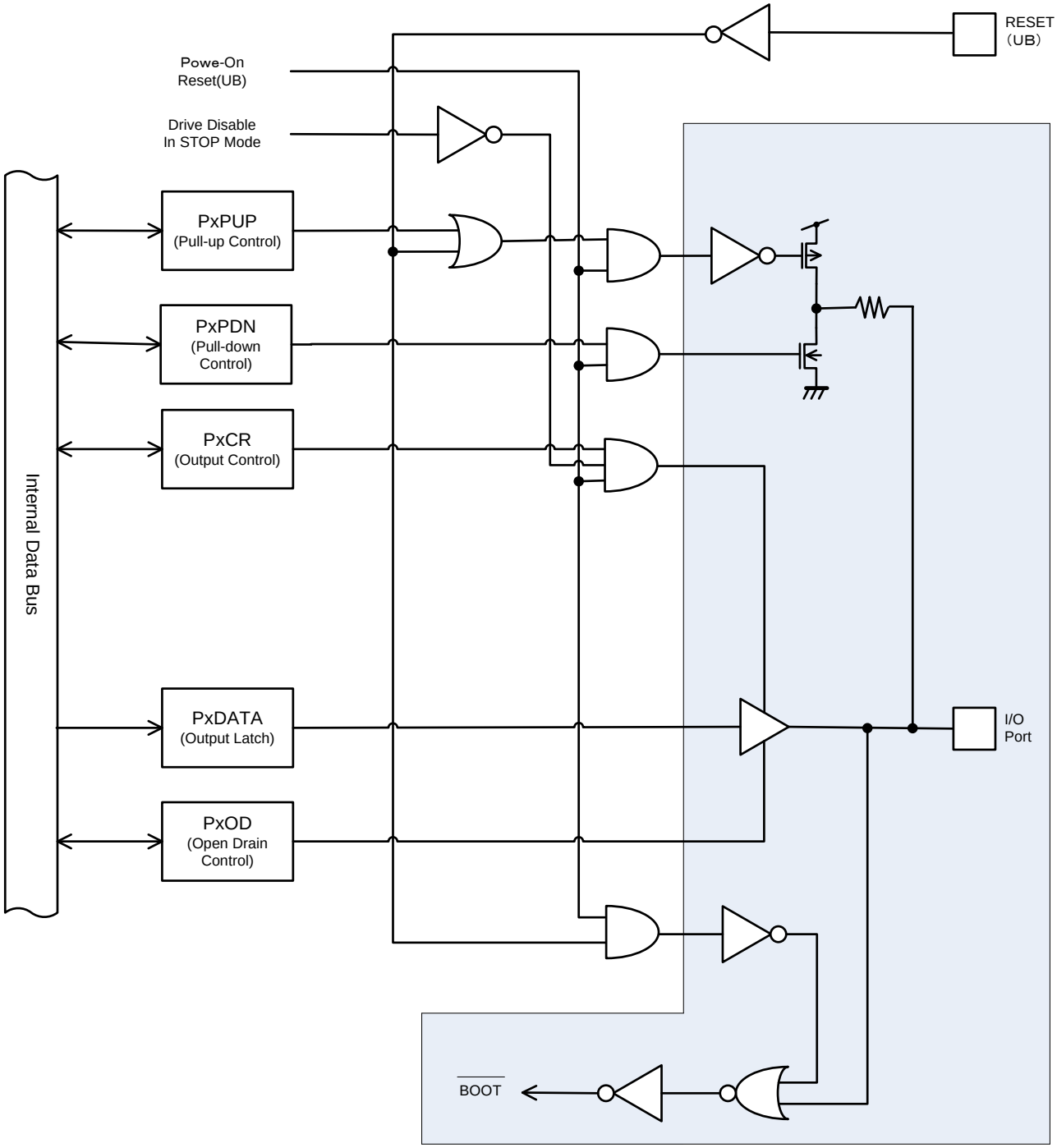


Figure 8-6 Port Type FT6

9 16-bit Timer/Event Counters (TMRBs)

Important
Neither TB3IN,TB3OUT,TB5IN nor TB5OUT are allocated in TMPM382 (64-pin version).
Please do not use the function (Up-counter source clock selection and capture operation that uses terminal TB3IN / TB5IN, Timer flip-flop output that uses terminal TB3OUT / TB5OUT.) to use a pertinent terminal.
However, TMRB3 and TMRB5 can be used by selecting a built-in clock as 16 bit interval timer.

9.1 Outline

TMPM380 have the eight channels multi-functional 16-bit timer/event counter. (TMRB0 through TMRB7) TMRBs operate in the following five operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square-wave output mode (PPG)
- External trigger programmable square-wave output mode (PPG)
- Timer synchronous mode (capable of setting output mode for each 4ch)

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

- One-shot pulse generation from an external trigger pulse
- Frequency measurement
- Pulse width measurement

9.2 Specification differences among channels

Channels (TMRB0 through TMRB7) operate independently and the functions are same except the differences as shown in Table 9-1 to Table 9-3. Therefore, the operational descriptions here are explained only for TMRB0.

The channels shown below are used as the capture or start trigger.

(1) The flip-flop output of TMRB 2, TMRB 5 and TMRB 7 can be used as the capture trigger of other channels.
- TB2OUT => available for TMRB 3 through TMRB 5
- TB5OUT => available for TMRB 6 through TMRB 7
- TB7OUT => available for TMRB 0 through TMRB 2

(2) The start trigger of the timer synchronous mode (with TBPRUN,TBRUN)
- TMRB0 => can start with TMRB 1 through TMRB 3 synchronously
- TMRB4 => can start with TMRB 5 through TMRB 7 synchronously

Table 9-1 Differences in the Specifications of TMRB Modules (1) for TMPM380

Specification / Channel	External pins		Trigger	
	External clock/ capture trigger input pins	Timer flip-flop output pin	Timer for capture triggers	Timer for synchronous start triggers
TMRB0	TB0IN	TB0OUT	TB7OUT	-
TMRB1	TB1IN	TB1OUT	TB7OUT	TB0PRUN,TB0RUN
TMRB2	TB2IN	TB2OUT	TB7OUT	TB0PRUN,TB0RUN
TMRB3	TB3IN	TB3OUT	TB2OUT	TB0PRUN,TB0RUN
TMRB4	TB4IN	TB4OUT	TB2OUT	-
TMRB5	TB5IN	TB5OUT	TB2OUT	TB4PRUN,TB4RUN
TMRB6	TB6IN	TB6OUT	TB5OUT	TB4PRUN,TB4RUN
TMRB7	TB7IN	TB7OUT	TB5OUT	TB4PRUN,TB4RUN

Table 9-2 Differences in the Specifications of TMRB Modules (2) for TMPM382

Specification / Channel	External pins		Trigger	
	External clock/ capture trigger input pins	Timer flip-flop output pin	Timer for capture triggers	Timer for synchronous start triggers
TMRB0	TB0IN	TB0OUT	TB7OUT	-
TMRB1	TB1IN	TB1OUT	TB7OUT	TB0PRUN,TB0RUN
TMRB2	TB2IN	TB2OUT	TB7OUT	TB0PRUN,TB0RUN
TMRB3	―	―	TB2OUT	TB0PRUN,TB0RUN
TMRB4	TB4IN	TB4OUT	TB2OUT	-
TMRB5	―	―	TB2OUT	TB4PRUN,TB4RUN
TMRB6	TB6IN	TB6OUT	TB5OUT	TB4PRUN,TB4RUN
TMRB7	TB7IN	TB7OUT	TB5OUT	TB4PRUN,TB4RUN

Table 9-3 Differences in the Specifications of TMRB Modules (3)

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

9.3 Configuration

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

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

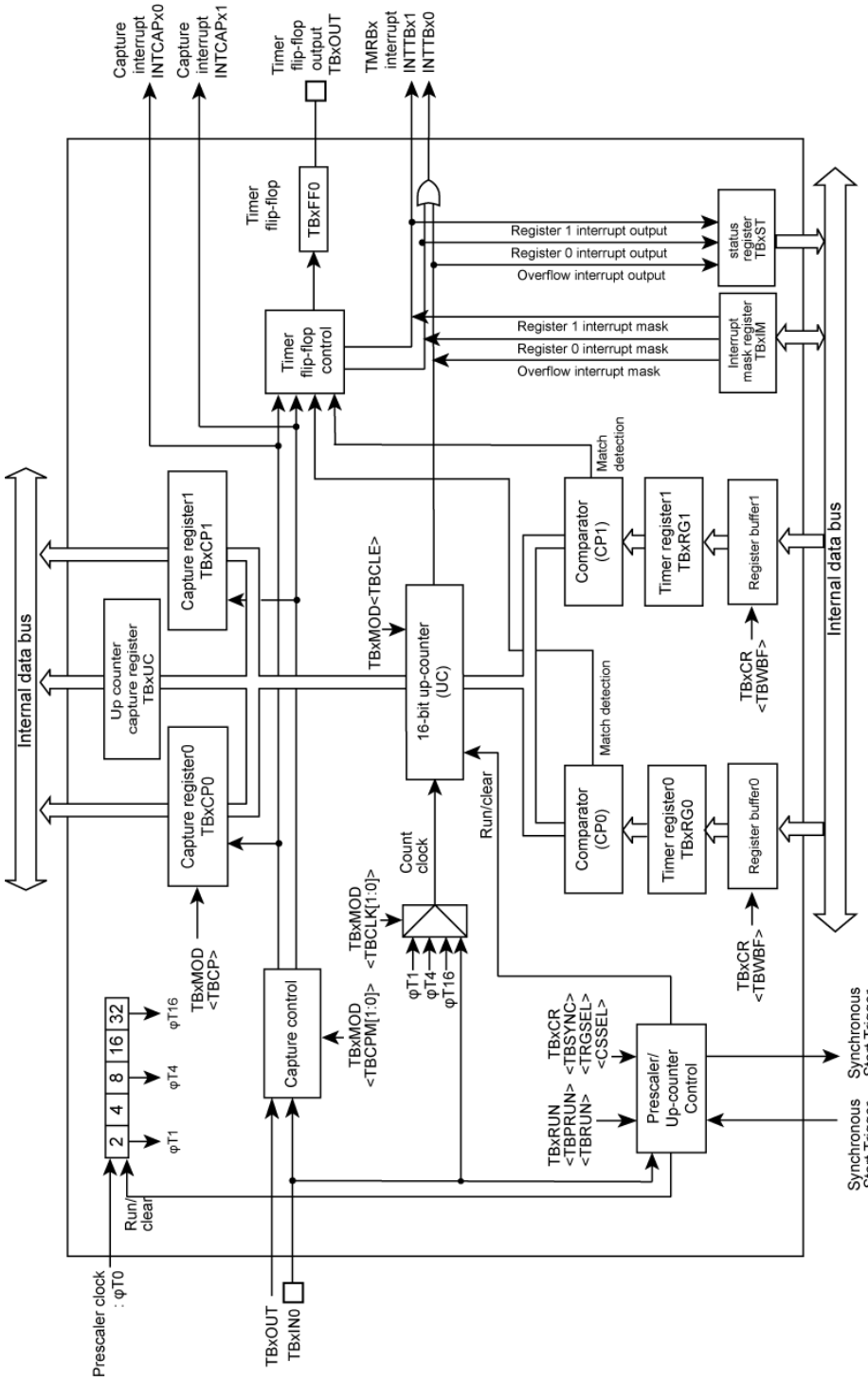


Fig 9-1 TMRB0 Block Diagram (the same applies to channels 1 through 7) for TMPM380, TMRB0 Block Diagram (the same applies to channels 0,1,2,4,6,7) for TMPM382

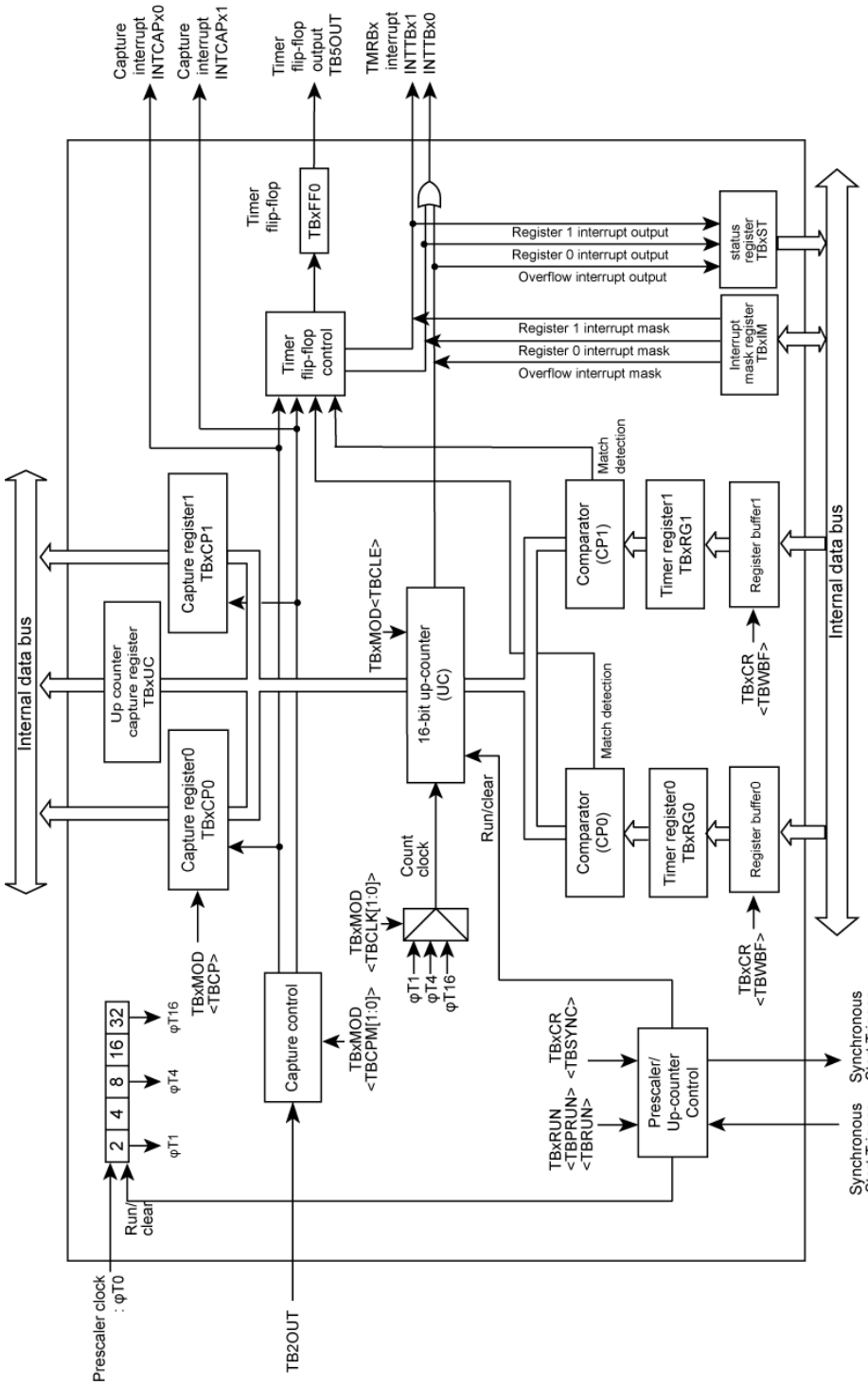


Fig 9-2 TMRB3 Block Diagram (the same applies to channel 5) for TMPM382

Because the TMRB3 and the TMRB5 of TMPM382 (64 pins) have no external input pins (TBxIN), the TB2OUT is the only capture trigger signal. The timer flip-flop output cannot be used as an external output, because it is not assigned to an external output. However, the TB5OUT can be used as a capture trigger signal for the TMRB6 and the TMRB7.

9.4 Registers

9.4.1 TMRB registers

Table 9-4 shows the register names and addresses of each channel.

Table 9-4 TMRB registers

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

Specification \ Channel		TMRB4		TMRB5		TMRB6		TMRB7	
Register names (addresses)	Timer enable register	TB4EN	0x4001_0100	TB5EN	0x4001_0140	TB6EN	0x4001_0180	TB7EN	0x4001_01C0
	Timer RUN register	TB4RUN	0x4001_0104	TB5RUN	0x4001_0144	TB6RUN	0x4001_0184	TB7RUN	0x4001_01C4
	Timer control register	TB4CR	0x4001_0108	TB5CR	0x4001_0148	TB6CR	0x4001_0188	TB7CR	0x4001_01C8
	Timer mode register	TB4MOD	0x4001_010C	TB5MOD	0x4001_014C	TB6MOD	0x4001_018C	TB7MOD	0x4001_01CC
	Timer flip-flop control register	TB4FFCR	0x4001_0110	TB5FFCR	0x4001_0150	TB6FFCR	0x4001_0190	TB7FFCR	0x4001_01D0
	Timer status register	TB4ST	0x4001_0114	TB5ST	0x4001_0154	TB6ST	0x4001_0194	TB7ST	0x4001_01D4
	Interrupt mask register	TB4IM	0x4001_0118	TB5IM	0x4001_0158	TB6IM	0x4001_0198	TB7IM	0x4001_01D8
	Timer up counter register	TB4UC	0x4001_011C	TB5UC	0x4001_015C	TB6UC	0x4001_019C	TB7UC	0x4001_01DC
	Timer register	TB4RG0 TB4RG1	0x4001_0120 0x4001_0124	TB5RG0 TB5RG1	0x4001_0160 0x4001_0164	TB6RG0 TB6RG1	0x4001_01A0 0x4001_01A4	TB7RG0 TB7RG1	0x4001_01E0 0x4001_01E4
	Capture register	TB4CP0 TB4CP1	0x4001_0128 0x4001_012C	TB5CP0 TB5CP1	0x4001_0168 0x4001_016C	TB6CP0 TB6CP1	0x4001_01A8 0x4001_01AC	TB7CP0 TB7CP1	0x4001_01E8 0x4001_01EC

TB3FFCR is only for TMPM380.

9.4.1.1 TMRBn enable register (channels 0 through 7)

TMRBn enable register (n=0~7)

TBnEN (0x4001_0xx0)

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

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

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

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

＜TBHALT＞: Specifies the control in HALT mode during debug mode.

0: Clock not stops in HALT mode

1: Clock stops in HALT mode

9.4.1.2 TMRB RUN register (channels 0 through 7)

TMRBn RUN register (n=0~7)

TBnRUN (0x4001_0xx4)

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

＜TBRUN＞ :Controls the TMRBn count operation.

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

1: Start counter

＜TBPRUN＞:Controls the TMRBn prescaler operation.

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

1: Start prescaler operation.

9.4.1.3 TMRB control register (channels 0 through 7)

TMRBn control register (n=0~7)

TBnCR (0x4001_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol	TBWBF		TBSYNC		I2TB		TRGSEL	CSSEL
Read/Write	R/W	R/W	R/W	R	R/W	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0: Disabled 1: Enabled	Write “0”.	Sync. mode select 0:Asyunc. 1:Syunc.	“0” is read.	During IDLE mode 0:Stop 1:Operation	“0” is read.	External Trigger 0: Rising edge 1: Falling edge	Counter Start 0: Software start 1: External trigger

<CSSEL>: Selects how the timer starts counting.

0: Select software for timer count start.

1: Select external trigger for timer count start.

Write “0” in case of using TMRB3 and TMRB5 of TMPM382.

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

0: Select rising edge of external trigger.

1: Select falling edge of external trigger.

Write "0" in case of using TMRB3 and TMRB5 of TMPM382.

＜I2TB＞:Controls the clock keep/ stop operation during the IDLE mode.

0: Stop the clock.

1: Keep clock operation during IDLE mode.

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

0: Disable Double Buffer.

1: Enable Double Buffer.

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

9.4.1.4 TMRB mode register (channels 0 thorough 7)

TMRBn mode register(n=0~7)

TBnMOD (0x4001_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol		TBRSWR	TBCP	TBCPM1	TBCPM0	TBCLE	TBCLK1	TBCLK0
Read/Write	R	R/W	W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Writes to timer registers 0 and 1 (when double buffering is enabled) 0: Can be written separately 1: Must be written simultaneously	Software capture control 0: Software capture 1: Don't care	Capture timing 00: Disable 01: TBnIN↑ 10: TBnIN↑ TBnIN↓ 11: TBnOUT↑ TBnOUT↓		Up-counter clear control 0:Clear and disable 1:Clear and enable	Source clock 00: TBnIN pin input 01: φT1 10: φT4 11: φT16	

<TBCLK1:0>:Selects the TMRBn timer count clock.

00: select TBnIN input pin

01: select φT1 (1/2φT0)

10: select φT4 (1/8φT0)

11: select φT16 (1/32φT0)

"00" is disable in case of using TMRB3 and TMRB5 of TMPM382.

<TBCLE>:Clears and controls the TMRBn up-counter.

"0": Disables clearing of the up-counter.

"1": Clears up-counter if there is a match with timer register 1 (TBnRG1).

＜TBCPM1:0＞:Specifies TMRBn capture timing.

"00": Capture disable

"01": Takes count values into capture register 0 (TBnCP0) upon rising of TBnIN pin input.

"10": Takes count values into capture register 0 (TBnCP0) upon rising of TBnIN pin input and into capture register 1 (TBnCP1) upon falling of TBnIN pin input.

"11": Takes count values into capture register 0 (TBnCP0) upon rising of 16-bit timer match output (TBnOUT) and into capture register 1 (TBnCP1) upon falling of TBnOUT.

(TMRB3 through TMRB5:TB2OUT, TMRB6 and TMRB7:TB5OUT, TMRB0 through and TMRB2:TB7OUT).

"01" and "10" are disable in case of using TMRB3 and TMRB5 of TMPM382.

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

<TBRSWR>: Controls the timing to write to timer registers 0 and 1 when double buffering is enabled.

"0": Timer registers 0 and 1 can be written separately, even in case writing preparation is ready for only one resister.

"1": In case both resisters are not ready to be written, Timer registers 0 and 1 can not be written

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

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

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

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

TBnFFCR (0x4001_0xx0)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol			TBC1T1	TBC0T1	TBE1T1	TBE0T1	TBFF0C1	TBFF0C0
Read/Write	R		R/W				R/W	
After reset	1	1	0	0	0	0	1	1
Function	"11" is read.		TBnFF0 reverse trigger 0: Disable 1: Enable				TBnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * These are always read as "11."	
			When the up-counter value is taken into TBnCP1	When the up-counter value is taken into TBnCP0	When the up-counter matches TBnRG1	When the up-counter matches TBnRG0		

＜TBFF0C1:0＞:Controls the timer flip-flop.

00 : Reverses the value of TBnFF0

01 : Sets TBnFF0 to "1".

10 : Clears TBnFF0 to "0".

11 :Don't care

＜TBE0T1＞:Reverses the timer flip-flop when the up-counter matches the timer register 0 (TBnRG0).

0: TBnFF0 not reverse

1: TBnFF0 reverse

＜TBE1T1＞:Reverses the timer flip-flop when the up-counter matches the timer register 1 (TBnRG1).

0: TBnFF0 not reverse

1: TBnFF0 reverse

＜TBC0T1＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 0 (TBnCP0).

0: TBnFF0 not reverse

1: TBnFF0 reverse

＜TBC1T1＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 1 (TBnCP1).

0: TBnFF0 not reverse
1: TBnFF0 reverse

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

TB3FFCR is only for TMPM380.

9.4.1.6 TMRB status register (channels 0 through 7)

TMRBn status register (n=0~7)

TBnST (0x4001_0xx4)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol						INTTBOF	INTTB1	INTTB0
Read/Write	R					R		
After reset	0					0	0	0
Function	"0" is read.					Interrupt for Overflow 0: Interrupt not generated 1: Interrupt generated	Interrupt for match with TBnREG1 0: Interrupt not generated 1: Interrupt generated	Interrupt for match with TBnREG0 0: Interrupt not generated 1: Interrupt generated

<INTTB0>:Interrupt generated status for a match with timer register 0 (TBnRG0)
0: No interrupt generated
1: Interrupt generated
<INTTB1>:Interrupt generated status for a match with timer register 1 (TBnRG1)
0: No interrupt generated
1: Interrupt generated
<INTTBOF>:Interrupt generated status for an up-counter overflow occurs
0: No interrupt generated
1: Interrupt generated

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

9.4.1.7 TMRB interrupt mask register (channels 0 through 7)

TMRBn interrupt mask register (n=0~7)

TBnIM
(0x4001_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol						TBIMOF	TBIM1	TBIM0
Read/Write	R					R/W		
After reset	0					0	0	0
Function	"0" is read					Interrupt mask for Overflow 0:No Interrupt mask 1: Interrupt is masked	Interrupt mask for match with TBnREG1 0:No Interrupt mask 1: Interrupt is masked	Interrupt mask for match with TBnREG0 0:No Interrupt mask 1: Interrupt is masked

＜TBIM0＞: Interrupt mask for a match with timer register 0 (TBnRG0)
0: No interrupt mask
1: Interrupt is masked
＜TBIM1＞:Interrupt mask for a match with timer register 1 (TBnRG1).
0: No interrupt mask
1: Interrupt is masked
＜TBIMOF＞:Interrupt mask for an up counter overflow.
0: No interrupt mask
1: Interrupt is masked

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

9.4.1.8 TMRB read capture register (channels 0 through 7)

TBnUC read capture register (n=0~7)

TBnUC (0x4001_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBUC15	TBUC14	TBUC13	TBUC12	TBUC11	TBUC10	TBUC9	TBUC8
Read/Write	R							
After reset	0							
Function	Data obtained by read capture: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBUC7	TBUC6	TBUC5	TBUC4	TBUC3	TBUC2	TBUC1	TBUC0
Read/Write	R							
After reset	0							
Function	Data obtained by read capture: 7-0 bit							

<UC15-0>：Captured Up-counter value.

9.4.1.9 TMRB timer register (channels 0 through 7)

TBnRG0 timer register (n=0~7)

TBnRG0
(0x4001_0xx0)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBRG015	TBRG014	TBRG013	TBRG012	TBRG011	TBRG010	TBRG09	TBRG08
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBRG07	TBRG06	TBRG05	TBG04	TBRG03	TBRG02	TBRG01	TBRG00
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

<TBRG015-000>：16bit Compare value0 with Up- counter

TBnRG1 timer register (n=0~7)

TBnRG1
(0x4001_0xx4)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBRG115	TBRG114	TBRG113	TBRG112	TBRG111	TBRG110	TBRG19	TBRG18
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBRG17	TBRG16	TBRG15	TBRG14	TBRG13	TBRG12	TBRG11	TBRG10
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

<TBRG115-100>：16bit Compare value1 with Up- counter

9.4.1.10 TMRB capture register (channels 0 through 7)

TBnCP0capture register (n=0~9)

TBnCP0 (0x4001_0xx8)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBCP015	TBCP014	TBCP013	TBCP012	TBCP011	TBCP010	TBCP09	TBCP08
Read/Write	R							
After reset	0							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBCP07	TBCP06	TBCP05	TBCP04	TBCP03	TBCP02	TBCP01	TBCP00
Read/Write	R							
After reset	0							
Function	Timer capture value: 7-0 bit data							

<TBCP015-000>: 16bit up-counter capture value0

TBnCP1 capture register (n=0~7)

TBnCP1 (0x4001_0xxC)

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	TBCP115	TBCP114	TBCP113	TBCP112	TBCP111	TBCP110	TBCP19	TBCP18
Read/Write	R							
After reset	0							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	TBCP17	TBCP16	TBCP15	TBCP14	TBCP13	TBCP12	TBCP11	TBCP10
Read/Write	R							
After reset	0							
Function	Timer capture value: 7-0 bit data							

<TBCP115-100>: 16bit up-counter capture value1

9.5 Description of Operations for Each Circuit

The channels operate in the same way, except for the differences in their specifications as shown in Table 9-1 to Table 9-3. Therefore, the operational descriptions here are only for channel 0.

9.5.1 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter UC0. The prescaler input clock ϕT0 is fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCLK2:0> 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 stop of a prescaler is set with TB0RUN<TBPRUN> where writing “1” starts counting and writing “0” clears and stops counting. Table 9-5 show prescaler output clock resolutions.

Table 9-5 Prescaler Output Clock Resolutions @fc = 40MHz

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

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

9.5.2 Up-counter (UC0)

UC0 is a 16-bit binary counter.

- Source clock

UC0 source clock, specified by TB0MOD<TB0CLK1:0>, can be selected from either three types ϕT1, ϕT4 and ϕT16 of prescaler output clock or the external clock of the TB0IN pin.

- Count start/ stop

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

- Timing to clear UC0

1) When a compare match is detected

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

2) When UC0 stops

UC0 stops counting and clears counter value if TB0RUN <TBRUN> = “0”.

- UC0 overflow

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

9.5.3 Timer registers (TB0RG0, TB0RG1)

TB0RG0 and TB0RG1 are registers for setting values to compare with up-counter values and two registers are built into each channel. If the comparator detects a match between a value set in this timer register and that in a UC0 up-counter, it outputs the match detection signal.

- Configuration

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

- Default setting

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

- Register setting

1) When not using double-buffering

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

2) When using double-buffering

TB0RG0/ TB0RG1 and the register buffer0/ register buffer1 are assigned to the same address. If <TBWBF> = “0,” the same value is written to TB0RG0, TB0RG1 and each register buffer; if <TBWBF> = “1,” the value is only written to each register buffer. Therefore, in order to write an initial value to the timer register, the register buffers must be set to “disable”. Then set <TBWBF> = “1”and write the following data to the register.

Note: The value of TB0RG0/1 must be set as TB0RG0 < TB0RG1 in PPG mode.

- Interrupt

INTTB00 is generated by UP0 count value matching with TB0RG0 value.
INTTB01 is generated by UP0 count value matching with TB0RG1 value.

9.5.4 Capture Control

This is a circuit that controls the timing to latch UC0 up-counter values into the TB0CP0 and TB0CP1 capture registers. The timing to latch data is specified by TB0MOD <TB0CPM1:0>.

Software can also be used to capture values from the UC0 up-counter into the capture register; specifically, UC0 values are taken into the TB0CP0 capture register each time “0” is written to TB0MOD<TBCP0>. To use this capability, the prescaler must be running (TB0RUN<TBPRUN> = “1”).

9.5.5 Capture Registers (TB0CP0, TB0CP1)

These are 16-bit registers for latching values from the UC0 up-counter. To read data from the capture register, use a 16-bit data transfer instruction or read in the order of low-order bits followed by high-order bits.

9.5.6 Up-counter capture register (TB0UC)

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

9.5.7 Comparators (CP0,CP1)

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

9.5.8 Timer Flip-flop (TB0FF0)

The timer flip-flop (TB0FF0) is reversed by a match signal from the comparator and a latch signal to the capture registers. It can be enabled or disabled to reverse by setting the TB0FFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>.

The value of TB0FF0 becomes undefined after a reset. The flip-flop can be reversed by writing “00” to TB0FFCR<TBFF0C1:0>. It can be set to “1” by writing “01,” and can be cleared to “0” by writing “10.”

The TB0FF0 value can be output to the timer output pin:TB0OUT (shared with PA0). To enable timer output, the port A related registers PACR and PAFR1 must be programmed beforehand.

9.5.9 Capture interrupt (INTCAP00, INTCAP01)

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

9.6 Description of Operations for Each Mode

9.6.1 16-bit Interval Timer Mode

-Generating interrupts at periodic cycles

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

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

9.6.2 16-bit Event Counter Mode

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

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

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

9.6.3 16-bit Programmable Square Wave Output Mode (PPG)

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

Programmable square waves can be output from the TB0OUT pin by triggering the timer flip-flop (TB0FF) to reverse when the set value of the up-counter (UCO) matches the set values of the timer registers (TB0RG0 and TB0RG1). Note that the set values of TB0RG0 and TB0RG1 must satisfy the following requirement:

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

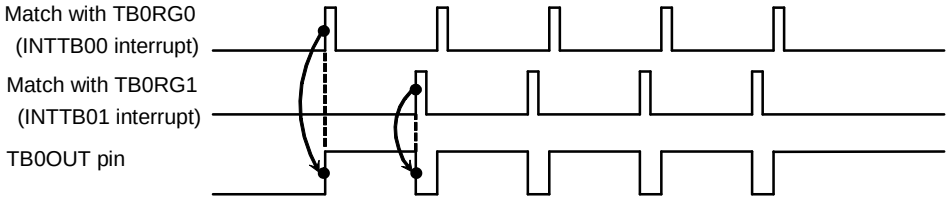


Fig 9-3 Example of Output of Programmable Square Wave (PPG)

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

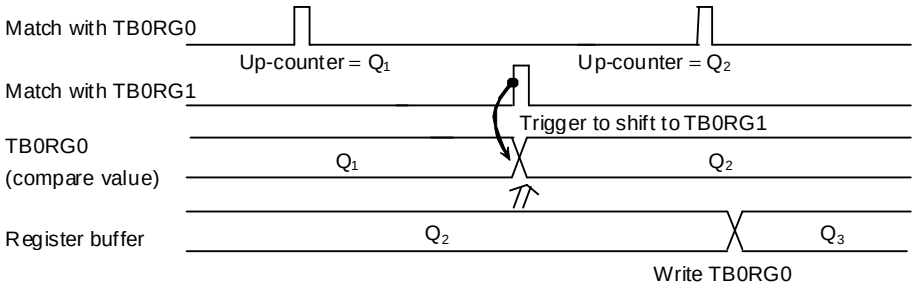


Fig 9-4 Register Buffer Operation

The block diagram of this mode is shown below.

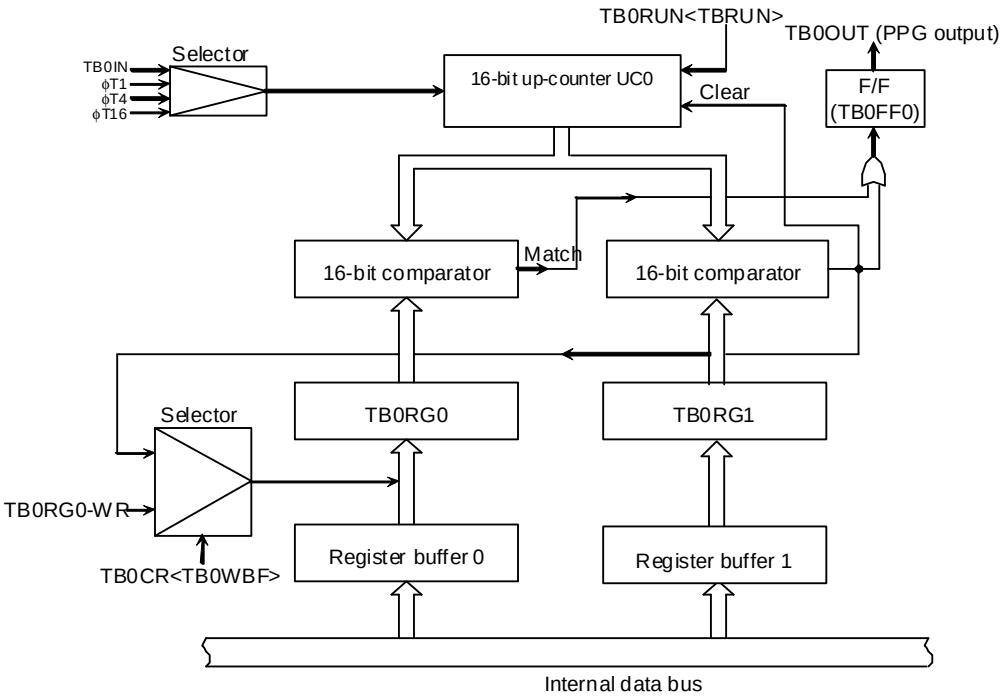


Fig 9-5 Block Diagram of 16-bit PPG Mode

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

		7	6	5	4	3	2	1	0	
TB0EN	←	1	X	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	X	X	X	X	X	0	X	0	Stops the TMRB0
TB0RG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits *32-bits register length)
	←	*	*	*	*	*	*	*	*	
TB0RG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits *32-bits register length)
	←	*	*	*	*	*	*	*	*	
TB0CR	←	1	0	X	0	0	0	0	0	Enables the TB0RG0 double buffering. (Changes the duty/cycle when the INTTB0 interrupt is generated)
TB0FFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TB0FF0 to reverse when a match with TB0RG0 or TB0RG1 is detected, and sets the initial value of TB0FF0 to "0."
TB0MOD	←	0	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function. (** = 01, 10, 11)
PACR	←	–	–	–	–	–	–	1	–	Assigns PA0 to output and TB0OUT
PAFR1	←	–	–	–	–	–	–	1	–	
TB0RUN	←	*	*	*	*	*	1	X	1	Starts TMRB0

X; Don't care –; no change

9.6.4 External trigger Programmable Square Wave Output Mode (PPG)

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

(1) The 16-bit up-counter (UC0) is programmed to count up on the rising edge of the TB0IN pin (TB0CR<TRGSEL0,CSSEL0>="01"). The TB0RG0 is loaded with the pulse delay (d), and the TB0RG1 is loaded with the sum of the TB0RG0 value (d) and the pulse width (P). The above settings must be done while the 16-bit up-counter is stopped (TB0RUN<TBRUN>=0).

(2) To enable the trigger for timer flip-flop, sets TB0FFCR<TBE1T1, TBE0T1> to 11. With this setting, the timer flip-flop reverses when 16-bit up-counter (UC0) corresponds to TB0RG0 or TB0RG1.

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

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

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

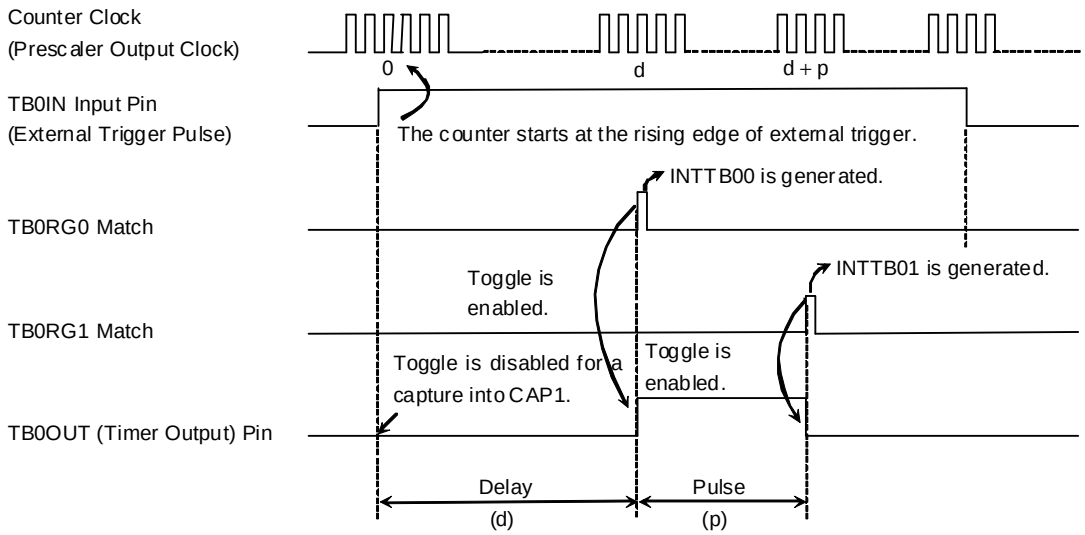


Fig. 9-5 One-shot pulse generation using an external count start trigger (with a delay)

9.6.5 Synchronous Timer Mode

Taking the synchronization of the start between timers by using the timer synchronous mode becomes possible.

It is possible to apply it to the drive such as motors by using the synchronous mode by the PPG output.

The synchronous mode is switched by TBxCR<TBSYNC >.

<TBSYNC>="0" : It operates according to the timing of each ch of the timer.

<TBSYNC>="1" : It outputs it synchronously.

It divides into two blocks of TMRB0,1,2,3 and TMRB4,5,6,7.

When I set <TBSYNC>="1", I will not start in timer start TBxRUN<TBPRUN, TBRUN>="1, 1" of each ch, and start synchronizing with TMRB0 and TMRB4.

Synchronous Timer Output Control Register (x=0~7)

TBxCR (0x4001_0xx8)

	7	6	5	4	3	2	1	0
bit Symbol	TBWBF		TBSYNC		I2TB		TRGSEL	CSSEL
Read/Write	R/W	R/W	R/W	R	R/W	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0: Disabled 1: Enabled	Write "0"	Sync. mode select 0:Asyunc 1:Syunc	"0" is read	During IDLE mode 0:Stop 1:Operation	"0" is read	External Trigger 0: Rising edge 1: Falling edge	Counter Start 0: Software start 1: External trigger

<I2TB> In the IDLE mode, it is possible to select it.

<TBSYNC> The timer output can be synchronized with each 4ch.

Please always set the TBSYNC bit to "0" about TMRB0,4 to have decided a synchronous start.

Please set the TBSYNC bit to "1" about the timer begun subordinating it.

<TBWBF> A double buffer of the timer register is controlled.

(note1) Please set the timer that outputs the same period to TBxRUN<TBPRUN and TBRUN>="1,1" before it starts by TMRB0,4.

(note2) Please set things except a synchronous output mode to TBxCR<TBSYNC>="0".
Other timers do not start until TMRB0,4 start when a synchronous output mode is set.

9.7 Application example using the Capture Function

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

(1) One-shot pulse output triggered by an external pulse

(2) Frequency measurement

(3) Pulse width measurement

(1) One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TB0IN pin. A trigger is generated at the rising of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TB0CP0).

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

TB0RG1 change must be completed before the next match.

In addition, the timer flip-flop control registers (TB0FFCR<TBE1T1, TBE0T1>) must be set to “11.” This enables triggering the timer flip-flop (TB0FF0) to reverse when UC0 matches TB0RG0 and TB0RG1. This trigger is disabled by the INTTB01 interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in Fig 9-6 One-shot Pulse Output (With Delay).”

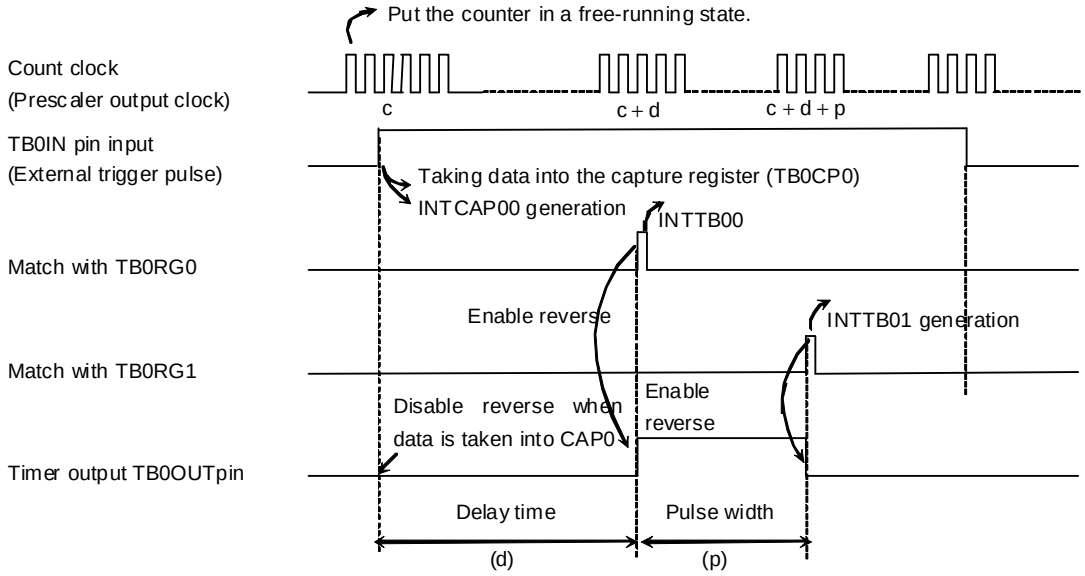


Fig 9-6 One-shot Pulse Output (With Delay)

Note: The value of TBnRG0/1 must be set as TBnRG0 < TBnRG1 in PPG mode.

If a delay is not required, TB0FF0 is reversed when data is taken into TB0CP0, and TB0RG1 is set to the sum of the TB0CPO value (c) and the one-shot pulse width (p), (c + p), by generating the INTTB00 interrupt. TB0RG1 change must be completed before the next match. TB0FF0 is enabled to reverse when UC0 matches with TB0RG1, and is disabled by generating the INTTB01 interrupt.

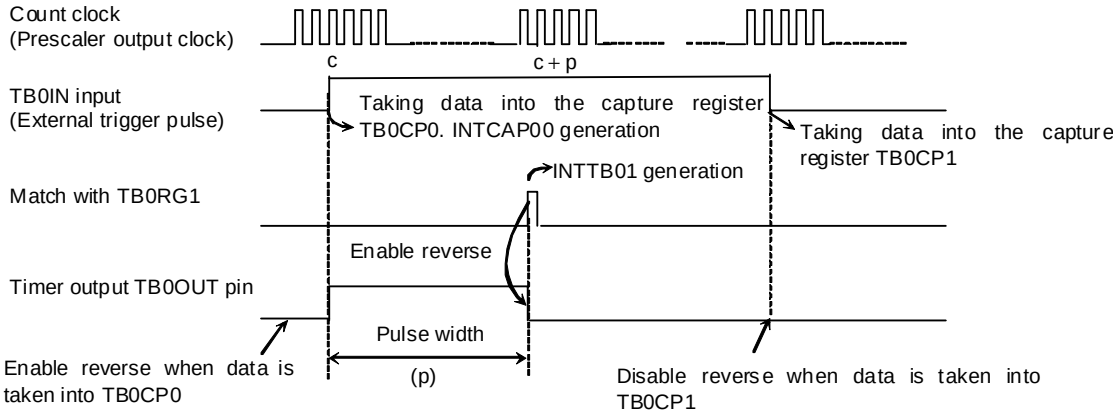


Fig 9-7 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

(2) Frequency measurement

The frequency of an external clock can be measured by using the capture function.

To measure frequency, another 16-bit timer is used in combination with the 16-bit event counter mode. As an example, we explain with TMRB0 and TMRB7. TB7OUT of the 16-bit timer TMRB7 is used to specify the measurement time.

The TB0IN pin input is selected as the TMRB0 count clock to perform the count operation using an external input clock. TB0MOD<TBCPM1:0> is set to “11.” This setting allows a count value of the 16-bit up-counter UC to be taken into the capture register (TB0CP0) upon rising of a timer flip-flop output (TB7OUT) of the 16-bit timer (TMRB7), and an UC counter value to be taken into the capture register (TB0CP1) upon falling of TB7OUT of the 16-bit timer (TMRB7).

A frequency is then obtained from the difference between TB0CP0 and TB0CP1 based on the measurement, by generating the INTTB70 and INTTB71 16-bit timer interrupt.

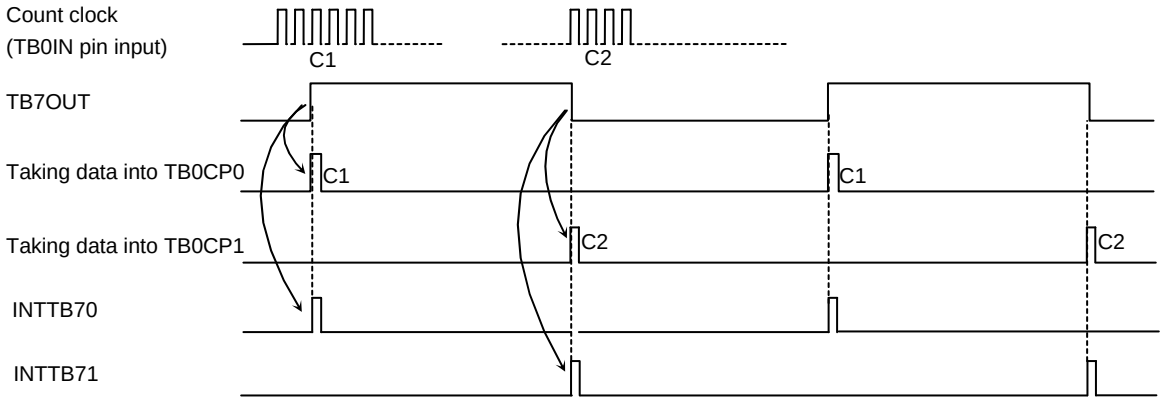


Fig 9-8 Frequency Measurement

For example, if the set width of TB7FF0 level “1” of the 16-bit timer is 0.5 s and if the difference between TB0CP0 and TB0CP1 is 100, the frequency is 100 / 0.5 s = 200 Hz.

(3) Pulse width measurement

By using the capture function, the “H” level width of an external pulse can be measured. Specifically, by putting it in a free-running state using the prescaler output clock, an external pulse is input through the TB0IN pin and the up-counter (UC0) is made to count up. A trigger is generated at each rising and falling edge of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TB0CP0, TB0CP1). The CPU must be programmed so that INTCAP01 is generated at the falling edge of an external pulse input through the TB0IN pin.

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

For example, if the difference between TB0CP0 and TB0CP1 is 100 and the cycle of the prescaler output clock is 0.5 us, the pulse width is 100×0.5 us $= 50$ us.

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

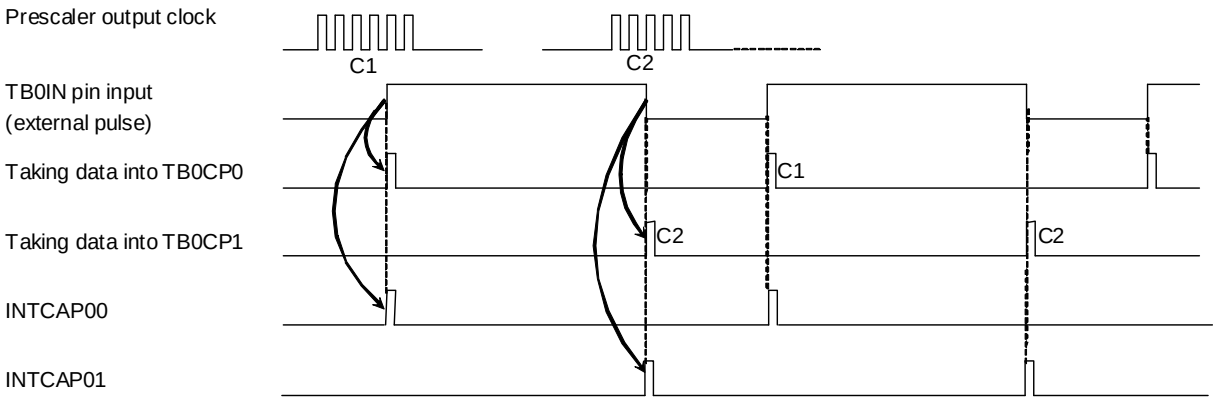


Fig 9-9 Pulse Width Measurement

The “L” level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCAP00 interrupt processing as shown in “Fig 9-9 Pulse Width Measurement” and this difference is multiplied by the cycle of the prescaler output clock to obtain the “L” level width.

10. 16-bit Multi-purpose Timers (MPTs)

Important
TMPM382 (64-pin version) does not implement MPT1 and MPT2.
Please do not use these functions if you use this product.

10.1 Outline

TMPM380 has three channels of 16-bit multi-purpose timer (MPT0 through MPT2). MPTs operate in the following operation modes:

＜Timer module>
16-bit interval timer mode
16-bit event counter mode
16-bit programmable square-wave output mode (PPG, Single wave)
Capture timer mode

＜IGBT Module>
16-bit programmable square-wave output mode (PPG, Two waves)
External trigger starting
Period matching detection function
Emergency stop function

＜PMD module>
3-phase motor control mode (PMD)

Note: PMD module is not available when using MPT2.

10.2 Specification differences among channels

Each channel (MPT0 through MPT2) operates independently and the functions are the same except the differences as shown in Table 10-1. Therefore, the operational descriptions here are explained only for MPT0.

Table 10-1 Differences in the Specifications of MPT Modules (1)

Specification / Channel	External pins					
	External clock/ capture trigger input pin	Timer flip-flop output pin	IGBT input pins	IGBT output pins	PMD input pin	PMD output pins
MPT0	MTTB0IN	MTTB0OUT	$\overline{\text{GEMG0}}$ MT0IN	MTOUT00 MTOUT10	$\overline{\text{EMG0}}$	UO0,VO0, WO0,XO0, YO0,ZO0
MPT1	MTTB1IN	MTTB1OUT	$\overline{\text{GEMG1}}$ MT1IN	MTOUT01 MTOUT11	$\overline{\text{EMG1}}$	UO1,VO1, WO1,XO1, YO1,ZO1
MPT2	MTTB2IN	MTTB2OUT	$\overline{\text{GEMG2}}$ MT2IN	MTOUT02 MTOUT12	–	–

10.3 Configuration

MPT consists of three modules including Timer (TMR), IGBT and PMD as shown below.

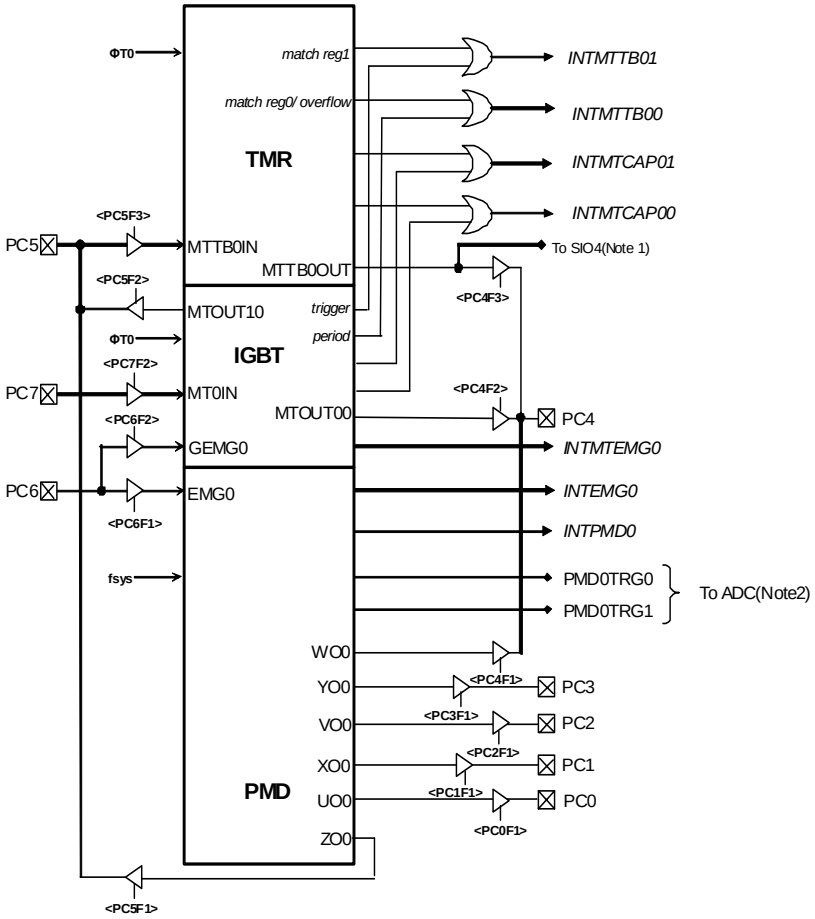


Figure 10-1 MPT0 Block Diagram (the same applies to channels 1 through2)

Note1:	When using MPT0 timer, Timer Flip-flop output (MTTB0OUT) can be selected as a serial transfer clock for SIO4 in the UART mode.
Note2:	PMD module is not available when using MPT2.

10.4 Registers

10.4.1 MPT registers

Table 10-2 shows the register names and addresses of each channel.

Table 10-2 MPT registers (1/3)

Specification \ Channel		MPT0		MPT1		MPT2	
Register names (addresses)	MPT enable register	MT0EN	0x4005_0800	MT1EN	0x4005_0880	MT2EN	0x4005_0900
	MPT RUN register	MT0RUN	0x4005_0804	MT1RUN	0x4005_0884	MT2RUN	0x4005_0904
	MPT control register	MT0TBCR	0x4005_0808	MT1TBCR	0x4005_0888	MT2TBCR	0x4005_0908
	MPT mode register	MT0TBMOD	0x4005_080C	MT1TBMOD	0x4005_088C	MT2TBMOD	0x4005_090C
	MPT flip-flop control register	MT0TBFFCR	0x4005_0810	MT1TBFFCR	0x4005_0890	MT2TBFFCR	0x4005_0910
	MPT status register	MT0TBST	0x4005_0814	MT1TBST	0x4005_0894	MT2TBST	0x4005_0914
	MPT Interrupt mask register	MT0TBIM	0x4005_0818	MT1TBIM	0x4005_0898	MT2TBIM	0x4005_0918
	MPT read capture register	MT0TBUC	0x4005_081C	MT1TBUC	0x4005_089C	MT2TBUC	0x4005_091C
	MPT Timer register	MT0RG0	0x4005_0820	MT1RG0	0x4005_08A0	MT2RG0	0x4005_0920
		MT0RG1	0x4005_0824	MT1RG1	0x4005_08A4	MT2RG1	0x4005_0924
	MPT Capture register	MT0CP0	0x4005_0828	MT1CP0	0x4005_08A8	MT2CP0	0x4005_0928
		MT0CP1	0x4005_082C	MT1CP1	0x4005_08AC	MT2CP1	0x4005_092C

Table 10-3 MPT registers (2/3)

Specification \ Channel		MPT0		MPT1		MPT2	
Register names (addresses)	IGBT control register	MT0IGCR	0x4005_0830	MT1IGCR	0x4005_08B0	MT2IGCR	0x4005_0930
	IGBT timer restart register	MT0IGRESTA	0x4005_0834	MT1IGRESTA	0x4005_08B4	MT2IGRESTA	0x4005_0934
	IGBT timer status register	MT0IGST	0x4005_0838	MT1IGST	0x4005_08B8	MT2IGST	0x4005_0938
	IGBT input control register	MT0IGICR	0x4005_083C	MT1IGICR	0x4005_08BC	MT2IGICR	0x4005_093C
	IGBT output control register	MT0IGOCR	0x4005_0840	MT1IGOCR	0x4005_08C0	MT2IGOCR	0x4005_0940
	IGBT timer register	MT0IGRG2	0x4005_0844	MT1IGRG2	0x4005_08C4	MT2IGRG2	0x4005_0944
		MT0IGRG3	0x4005_0848	MT1IGRG3	0x4005_08C8	MT2IGRG3	0x4005_0948
		MT0IGRG4	0x4005_084C	MT1IGRG4	0x4005_08CC	MT2IGRG4	0x4005_094C
	IGBT EMG control register	MT0IGEMGCR	0x4005_0850	MT1IGEMGCR	0x4005_08D0	MT2IGEMGCR	0x4005_0950
	IGBT EMG status register	MT0IGEMGST	0x4005_0854	MT1IGEMGST	0x4005_08D4	MT2IGEMGST	0x4005_0954

Table 10-4 MPT registers (3/3)

Specification \ Channel		MPT0		MPT1		MPT2
Register names (addresses)	PMD Enable Register	MTPD0MDEN	0x4005_0400	MTPD1MDEN	0x4005_0480	NA
	Port Output Mode Register	MTPD0PORTMD	0x4005_0404	MTPD1PORTMD	0x4005_0484	NA
	PMD Control Register	MTPD0MDCR	0x4005_0408	MTPD1MDCR	0x4005_0488	NA
	PWM Counter Status Register	MTPD0CNTSTA	0x4005_040C	MTPD1CNTSTA	0x4005_048C	NA
	PWM Counter Register	MTPD0MDCNT	0x4005_0410	MTPD1MDCNT	0x4005_0490	NA
	PWM Period Register	MTPD0MDPRD	0x4005_0414	MTPD1MDPRD	0x4005_0494	NA
	PMD Compare U Register	MTPD0CMPU	0x4005_0418	MTPD1CMPU	0x4005_0498	NA
	PMD Compare V Register	MTPD0CMPV	0x4005_041C	MTPD1CMPV	0x4005_049C	NA
	PMD Compare W Register	MTPD0CMPW	0x4005_0420	MTPD1CMPW0	0x4005_04A0	NA
	PMD Output Control Register	MTPD0MDOUT	0x4005_0428	MTPD1MDOUT	0x4005_04A8	NA
	PMD Output Setting Register	MTPD0MDPOT	0x4005_042C	MTPD1MDPOT	0x4005_04AC	NA
	EMG Release Register	MTPD0EMGREL	0x4005_0430	MTPD1EMGREL	0x4005_04B0	NA
	EMG Control Register	MTPD0EMGCR	0x4005_0434	MTPD1EMGCR	0x4005_04B4	NA
	EMG Status Register	MTPD0EMGST	0x4005_0438	MTPD1EMGST	0x4005_04B8	NA
	Dead Time Register	MTPD0DTR	0x4005_0444	MTPD1DTR	0x4005_04C4	NA
	Trigger Compare 0 Register	MTPD0TRGCMP0	0x4005_0448	MTPD1TRGCMP0	0x4005_04C8	NA
	Trigger Compare 1 Register	MTPD0TRGCMP1	0x4005_044C	MTPD1TRGCMP1	0x4005_04CC	NA
	Trigger Control Register	MTPD0TRGCR	0x4005_0458	MTPD1TRGCR	0x4005_04D8	NA
	Trigger Output Mode Setting Register	MTPD0TRGMD	0x4005_045C	MTPD1TRGMD	0x4005_04DC	NA

NA: Not Available.

10.4.1.1 Operating Precautions for MPTs

Note1 : Do not change settings as following registers during timer is in operation (MTnRUN<MTRUN>="1").
MTnTBFFCR
MTnTBMOD
MTnTBCR
MTnIGCR
MTnIGICR
MTnIGEMGCR

Note2 : Setting of MTnRG0/1 must be set as "Value of MTnRG0 < Value of MTnRG1".

Note3 : Setting of MTnRG0/1/2/3/4 must be set as follows;
0 < Value of MTnRG0 < Value of MTnRG1 $\leqq$ Value of MTnIGRG4 $\leqq$ 0xffff
0 < Value of MTnIGRG2 < Value of MTnIGRG3 $\leqq$ Value of MTnIGRG4 $\leqq$ 0xffff

Note4 : Write to MTnRG0/1/2/3/4 by unit of 16 bits or 32 bits. Writing by units of bytes is prohibited.

Note5: Regardless of the timer operating or stopping, MTOUTn0 and MTOUTn1 output pins change according to a setting register of IGBT output control register (MTnIGOCR). Confirm the operating conditions beforehand when changing the MTnIGOCR setting.

Note6: During timer in operation, if IGBT timer restart register MTnIGRESTA<IGRESTA> is set to "1", timer counter can be cleared and restarted. Confirm the operating conditions beforehand when changing the MTnIGRESTA setting.

Note7: When changing the noise eliminating time setting of IGBT timer trigger start pin (MTnTBIN) specified with MTnIGCR<IGNCSEL3:0>, timer counter must be stopped. When changing the setting of the trigger start pin (MTnTBIN) of IGBT timer, the timer must be waited to start (MTnRUN<MTRUN>="1") until noise elimination time (more than specified time for noise elimination) passes after changing setting of noise elimination time.

Note8: When changing the noise eliminating setting of GEMGn pin specified with MTnIGEMGCR<IGEMGCNT[3:0]>, EMG protection circuit must be disabled (MTnIGEMGCR<IGEMGEN>="0").

Note9: If MTnIGCR<IGSTP> is set to "10" (stop counter after completing output in the current period) and stopped by MTnRUN<MTRUN>, confirm a period interrupt generation before changing the timer setting and restart.

Note10: While timer is stopping, a value of timer counters (MTUCn) is not captured. Since data in the capture register (MTnCP0, MTnCP1) is maintained, a preceding captured value of timer counter can be read.

10.4.1.2 MPTn enable register <Common>

MTnEN (n=0,1,2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	MTEN	MTHALT	-	-	-	-	-	MTMODE
Read/Write	R/W	R/W	R					R/W
After reset	0	0	0					0
Function	MPT operation 0: Disabled 1: Enabled	Control in Core HALT at debug 0: Disabled 1: Enabled	"0" is read.					Operation mode 0: Timer 1: IGBT

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

＜MTHALT＞: Specifies the control in Core HALT during debug mode.

[In Timer mode]

0: Clock not stops

1: Clock stops

[In IGBT mode]

0: Clock does not stop and not control MTOUT0n/MTOUT1n

1: Clock stops and control MTOUT0n/MTOUT1n depend on MTnIGEMGCR<IGEMGCOC> setting

＜MTMODE＞: Select operation mode

0: Timer mode

1: IGBT mode

To use the MPT, enable the MPT operation (<MTEN>="1") before programming each register in the MPT module.

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

10.4.1.3 MPT RUN register <Common>

MTnRUN (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	MTPRUN	-	MTRUN
Read/Write	R					R/W	R	R/W
After reset	0					0	0	0
Function	"0" is read.					Prescaler Run/Stop Control 0: stop and clear 1: Count	"0" is read.	MTP Run/Stop Control 0: stop and clear 1: Count

＜MTRUN＞ :Controls the MPTn counting operation.

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

1: Start counter

＜MTPRUN＞:Controls the MPTn prescaler operation.

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

1: Start prescaler operation.

10.4.1.4 MPT control register <Timer mode>

MTnTBCR (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit Symbol	MTTBWBF	-	-	-	MTI2TB	-	**MTTBTRGSEL**	MTTBCSSEL
Read/Write	R/W	R/W	R/W	R	R/W	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer control 0: Disabled 1: Enabled	Write “0”.	Write “0”.	“0” is read.	Operation in IDLE mode 0:Stop 1:Operation	“0” is read.	External Trigger selection 0: Rising edge 1: Falling edge	Counter Start selection 0: Software start 1: External trigger

<MTTBCSSEL>: Selects how the timer starts counting.

0: Select software for timer count start.

1: Select external trigger for timer count start.

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

0: Select rising edge of external trigger.

1: Select falling edge of external trigger.

<MTI2TB>: Controls the clock suply to keep/stop in the IDLE mode.

0: Stop the clock.

1: Keep clock operation during IDLE mode.

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

0: Disable Double Buffer.

1: Enable Double Buffer.

(Note1) MTnTBCR resister must not be changed during Timer in operation (MTnRUN<MTRUN>=”1”)

(Note2) Regardless of the <MTTBWBF> setting, double buffer is enabled automatically during IGBT mode.

10.4.1.5 MPT mode register <Timer mode>

MTnTBMOD (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	-	**MTTBRSWR**	MTTBCP	MTTBCPM1	MTTBCPM0	MTTBCLE	MTTBCLK1	MTTBCLK0
Read/Write	R	R/W	W	R/W		R/W	R/W	
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Writes to timer registers 0 and 1 **(when MTTBWBF=1)** 0: Can be written separately 1: Must be written simultaneously	Software capture control 0: Software capture 1: Don't care	Capture timing 00: Disable 01: MTnTBIN ↑ 10: MTnTBIN ↑ MTnTBIN ↓ 11: Disable		Up-counter control 0:disable Clearing 1: enable clearing	Source clock for Timer 00: MTnTBIN pin input 01: φT1 10: φT4 11: φT16	

＜MTTBCLK1:0＞:Selects the MPT timer count source clock.

00: select MTnTBIN input pin

01: select φT1 (1/2φT0)

10: select φT4 (1/8φT0)

11: select φT16 (1/32φT0)

＜MTTBCLE＞:Clears and controls the MPTn up-counter.

"0": Disables clearing of the up-counter.

"1": Clears up-counter if there is a match with timer register 1 (MTnRG1).

＜MTTBCPM1:0＞:Specifies MPTn capture timing.

"00": Capture disable

"01": Takes count values into capture register 0 (MTnCP0) upon rising of MTnTBIN pin input.

"10": Takes count values into capture register 0 (MTnCP0) upon rising of MTnTBIN pin input. Takes count values into capture register 1 (MTnCP1) upon falling of MTnTBIN pin input.

"11": Capture disable

＜MTTBCP＞:Captures count values by software and takes them into capture register 0 (MTnCP0).

<MTTBRSWR>: Controls the timing to write to timer registers 0 and 1 when double buffering is enabled.

"0": Timer registers 0 and 1 can be written separately, even in case writing preparation is ready for only one resister.

"1": In case both resisters are not ready to be written, Timer registers 0 and 1 cannot be written.

(Note 1) The value read from MTnTBMOD<MTTBCP> is "1".

(Note 2) MTnMOD register must not be changed during Timer in operation (MTnRUN<MTRUN>="1")

10.4.1.6 MPT flip-flop control register <Timer mode>

MTnTBFFCR (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	**MTTBC1T1**	**MTTBC0T1**	**MTTBE1T1**	**MTTBE0T1**	**MTTBFF0C1**	**MTTBFF0C0**
Read/Write	R		R/W				R/W	
After reset	1	1	0	0	0	0	0	0
Function	"11" is read.		MTnFF0 reverse trigger control 0: Disable trigger 1: Enable trigger				MTnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * This is always read as "11."	
			When the up-counter value is taken into MTnCP1	When the up-counter value is taken into MTnCP0	When the up-counter matches MTnRG1	When the up-counter matches MTnRG0		

＜MTTBFF0C1:0＞:Controls the timer flip-flop.

00 : Reverses the value of MTnFF0
01 : Sets MTnFF0 to "1".
10 : Clears MTnFF0 to "0".
11 :Don't care

＜MTTBE0T1＞:Reverses the timer flip-flop when the up-counter matches the timer register 0 (MTnRG0).

0: MTnFF0 not reverse
1: MTnFF0 reverse

＜MTTBE1T1＞:Reverses the timer flip-flop when the up-counter matches the timer register 1 (MTnRG1).

0: MTnFF0 not reverse
1: MTnFF0 reverse

＜MTTBC0T1＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 0 (MTnCP0).

0: MTnFF0 not reverse
1: MTnFF0 reverse

＜MTTBC1T1＞:Reverses the timer flip-flop when the up-counter value is taken into the capture register 1 (MTnCP1).

0: MTnFF0 not reverse

1: MTnFF0 reverse

(Note) MTnTBFFCR register must not be changed during Timer in operation (MTnRUN <MTRUN>="1").

10.4.1.7 MPT status register <Timer mode>

MTnTBST (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	**MTTBINTTBOF**	MTTBINTTB1	MTTBINTTB0
Read/Write	R					R		
After reset	0					0	0	0
Function	"0" is read.					0: No interrupt generated 1:Interrupt generated	0: No interrupt generated 1:Interrupt generated	0: No interrupt generated 1:Interrupt generated

<MTTBINT0>:Interrupt generated status for a match with timer register 0 (MTnRG0)
0: No interrupt generated
1: Interrupt generated (INTMTnTB0)

<MTTBINT1>:Interrupt generated status for a match with timer register 1 (MTnRG1)
0: No interrupt generated
1: Interrupt generated (INTMTnTB1)

<MTTBINTOF>:Interrupt generated status for an up-counter overflow occurs
0: No interrupt generated
1: Interrupt generated (INTMTnTB0)

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

10.4.1.8 MPT interrupt mask register <Timer mode>

MTnTBIM (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	MTTBIMOF	MTTBIM1	MTTBIM0
Read/Write	R					R/W		
After reset	0					0	0	0
Function	"0" is read					0: No Interrupt mask 1: Interrupt is masked	0: No Interrupt mask 1: Interrupt is masked	0: No Interrupt mask 1: Interrupt is masked

＜MTTBIM0＞: Interrupt mask for a match with timer register 0 (MTnRG0)
0: No interrupt mask
1: Interrupt is masked

＜MTTBIM1＞:Interrupt mask for a match with timer register 1 (MTnRG1).
0: No interrupt mask
1: Interrupt is masked

＜MTTBIMOF＞:Interrupt mask for an up counter overflow.
0: No interrupt mask
1: Interrupt is masked

(Note) Even in case MTnTBIM set interrupt Mask, MTnTBST register has an interrupt requested status.

10.4.1.9 MPT read capture register <Timer mode>

MTnTBUC (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	MTUC15	MTUC14	MTUC13	MTUC12	MTUC11	MTUC10	MTUC9	MTUC8
Read/Write	R							
After reset	0							
Function	Data obtained by read capture: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	MTUC7	MTUC6	MTUC5	MTUC4	MTUC3	MTUC2	MTUC1	MTUC0
Read/Write	R							
After reset	0							
Function	Data obtained by read capture: 7-0 bit data							

<UC15-0>：Captured Up-counter value.

10.4.1.10 MPT timer register <Common>

MTnRG0 (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	MTRG015	MTRG014	MTRG013	MTRG012	MTRG011	MTRG010	MTRG09	MTRG08
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	MTRG07	MTRG06	MTRG05	MTRG04	MTRG03	MTRG02	MTRG01	MTRG00
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

<MTRG0[15:0]>：16bit Compare value0 with Up- counter

[Timer mode]
When Up-counter value match with MTnRG0[15:0], INTMTTBn0 is generated. And it can use to invert TMTBnOUT output.
[IGBT mode]
When Up-counter value match with MTnRG0[15:0], MTOUT0n output is changed to active level.

Note: Write to this register by unit of 16 bits or 32 bits. Writing by unit of 8 bits is prohibited.

MTnRG1 (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	MTRG115	MTRG114	MTRG113	MTRG112	MTRG111	MTRG110	MTRG19	MTRG18
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	MTRG17	MTRG16	MTRG15	MTRG14	MTRG13	MTRG12	MTRG11	MTRG10
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							

<MTRG1[15:0]>：16bit Compare value1 with Up- counter

[Timer mode]
When Up-counter value match with MTnRG1[15:0], INTMTTBn1 is generated. And it can use to invert MTnTBOUT output.
[IGBT mode]
When Up-counter value match with MTnRG1[15:0], MTOUT0n output is changed to inactive level.

Note: Write to this register by unit of 16 bits or 32 bits. Writing by unit of 8 bits is prohibited.

10.4.1.11 MPT capture register <Common>

MTnCP0 (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	MTCP015	MTCP014	MTCP013	MTCP012	MTCP011	MTCP010	MTCP09	MTCP08
Read/Write	R							
After reset	0							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	MTCP07	MTCP06	MTCP05	MTCP04	MTCP03	MTCP02	MTCP01	MTCP00
Read/Write	R							
After reset	0							
Function	Timer capture value: 7-0 bit data							

<MTCP0[15:0]>: 16 bit up-counter capture value 0

MTnCP1 (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	MTCP115	MTCP114	MTCP113	MTCP112	MTCP111	MTCP110	MTCP19	MTCP18
Read/Write	R							
After reset	0							
Function	Timer capture value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	MTCP17	MTCP16	MTCP15	MTCP14	MTCP13	MTCP12	MTCP11	MTCP10
Read/Write	R							
After reset	0							
Function	Timer capture value: 7-0 bit data							

<MTCP1[15:0]>: 16 bit up-counter capture value 1

10.4.1.12 IGBT control register <IGBT mode>

MTnIGCR (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	“0” is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	“0” is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	IGIDIS	IGPRD1	IGPRD0
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	“0” is read					Interrupt control when command start 0: enable 1: disable	Interrupt period selection 00: Every one- period 01: Every two- period 10: Every four- period 11: Reserved	
	7	6	5	4	3	2	1	0
bit Symbol	-	IGSNGL	IGSTP1	IGSTP0	IGSTA1	IGSTA0	IGCLK1	IGCLK0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	“0” is read	IGBT operation mode 0: Continuous output 1: One-time output	Select the state when stopped. 00: Immediately stop a with output initialized 01: Immediately stop and clear counter with output maintained 10: Stop counter after completing output in the current period. 11: Reserved		Start mode selection 00: command start and trigger capture mode 01: command start and trigger start mode 10: trigger start mode 11: Reserved		Source clock for IGBT 00: φT0 01: φT1 10: φT2 11: φT4	

<IGIDIS>: Interrupt control when command start
"0": enable
"1": disable

<IGPRD1:0>: Interrupt period selection
"00": Every one-period
"01": Every two-period
"10": Every four-period
"11": Reserved

<IGSNGL>: IGBT operation mode
"0": Continuous output
"1": One- time output

<IGSTP1:0>: Select the state when stopped.
"00": Immediately stop and counter with output initialized
"01": Immediately stop and clear counter with output maintained
"10": Stop counter after completing output in the current period
"11": Reserved

<IGSTA1:0>: Start mode selection
"00": command start and trigger capture mode
"01": command start and trigger start mode
"10": trigger start mode
"11": Reserved

＜IGCLK1:0>: Source clock selection for IGBT
"00": select φT0 (1/1φT0)
"01": select φT1 (1/2φT0)
"10": select φT2 (1/4φT0)
"11": select φT4 (1/8φT0)

(Note) MTnIGCR register must not be changed during Timer in operation (MTnRUN<MTRUN>="1").

10.4.1.13 IGBT timer restart register <IGBT mode>

MTnIGRESTA (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	IGRESTA
Read/Write	R	R	R	R	R	R	R	W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							Count restart control 0: Don't care 1: Restart ("0" is read.)

<IGRESTA>: Count restart control
"0": don't care
"1": Restart (counter clear & start)

10.4.1.14 IGBT timer status register 2, 3, 4 <IGBT mode>

MTnIGST (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	"0" is read.							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	IGST
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							Counter status 0: Stop 1: Operating

<IGST>: Counter status
"0": Stop
"1": Operating

10.4.1.15 IGBT input control register <IGBT mode>

MTnIGICR (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	IGTRGM	IGTRGSEL	-	-	IGNCSEL3	IGNCSEL2	IGNCSEL1	IGNCSEL0
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Trigger edge acceptance mode 0: Always accept trigger edges 1: Do not accept trigger edges during active output	Select trigger start edge 0: Start on trigger rising edge 1: Start on trigger falling edge	"0" is read		Select the duration of noise elimination for input 0000: No elimination (Note) 0001: eliminate pulses shorter than 16 / fsys[s] 0010: eliminate pulses shorter than 32 / fsys[s] : 1111: eliminate pulses shorter than 240 / fsys[s]			

<IGTRGM>: Trigger edge acceptance mode
"0": Always accept trigger edges
"1": Do not accept trigger edges during active output

<IGTRGSEL>: Select trigger start edge/ active level selection
"0": Start on trigger rising edge/ "High" level active
"1": Start on trigger falling edge/ "Low" level active

<IGNCSEL3:0>: Select the duration of noise elimination for input
The noise elimination time is calculated in the formula shown in below.
IGNCSEL[3:0]×16 / fsys

0000: No elimination
0001: eliminate pulses shorter than 16 / fsys[s]
0010: eliminate pulses shorter than 32 / fsys[s]
0011: eliminate pulses shorter than 48 / fsys[s]
0100: eliminate pulses shorter than 64 / fsys[s]
0101: eliminate pulses shorter than 80 / fsys[s]
0110: eliminate pulses shorter than 96 / fsys[s]
0111: eliminate pulses shorter than 112 / fsys[s]
1000: eliminate pulses shorter than 128 / fsys[s]
1001: eliminate pulses shorter than 144 / fsys[s]
1010: eliminate pulses shorter than 160 / fsys[s]
1011: eliminate pulses shorter than 176 / fsys[s]
1100: eliminate pulses shorter than 192 / fsys[s]
1101: eliminate pulses shorter than 208 / fsys[s]
1110: eliminate pulses shorter than 224 / fsys[s]
1111: eliminate pulses shorter than 240 / fsys[s]

(Note) MTnIGICR register must not be changed during the timer in operation (MTnRUN<MTRUN>="1").

10.4.1.16 IGBT output control register <IGBT mode>

MTnIGOCR (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	IGPOL1	IGPOL0	-	-	IGOEN1	IGOEN0
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read		Specify initial value of MTOUT1n output 0: Low 1: High	Specify initial value of MTOUT0n output 0: Low 1: High	"0" is read		MTOUT1n output control 0: Disable 1: Enable	MTOUT0n output control 0: Disable 1: Enable

<IGPOL1>: initial value of MTOUT1n output
"0": Low
"1": High
<IGPOL0>: initial value of MTOUT0n output
"0": Low
"1": High
<IGOEN1>: MTOUT1n output control
"0": Disable
"1": Enable
<IGOEN0>: MTOUT0n output control
"0": Disable
"1": Enable

10.4.1.17 IGBT timer register <IGBT mode>

MTnIGRG2 (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	IGRG215	IGRG214	IGRG213	IGRG212	IGRG211	IGRG210	IGRG29	IGRG28
Read/Write	R/W							
After reset	0							
Function	IGBT Timer value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	IGRG27	IGRG26	IGRG25	IGRG24	IGRG23	IGRG22	IGRG21	IGRG20
Read/Write	R/W							
After reset	0							
Function	IGBT Timer value: 7-0 bit data							

<IGRG2[15:0]>: When match with Up-counter, MTOUT1n is changed to active level.

Note1: Setting Value: MTnIGRG2<MTnIGRG3<MTnIGRG4.
Note2: Write to this register by unit of 16 bits or 32 bits. Writing by unit of 8 bits is prohibited.

MTnIGRG3 (n=0,1,2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	IGRG315	IGRG314	IGRG313	IGRG312	IGRG311	IGRG310	IGRG39	IGRG38
Read/Write	R/W							
After reset	0							
Function	IGBT Timer value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	IGRG37	IGRG36	IGRG35	IGRG34	IGRG33	IGRG32	IGRG31	IGRG30
Read/Write	R/W							
After reset	0							
Function	IGBT Timer value: 7-0 bit data							

<IGRG3[15:0]>: When match with Up-counter, MTOUT1n change to inactive level.

Note1: Setting Value: MTnIGRG2<MTnIGRG3<MTnIGRG4.
Note2: Write to this register by unit of 16 bits or 32 bits. Writing by unit of 8 bits is prohibited.

MTnIGRG4 (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	IGRG415	IGRG414	IGRG413	IGRG412	IGRG411	IGRG410	IGRG49	IGRG48
Read/Write	R/W							
After reset	0							
Function	IGBT Timer value: 15-8 bit data							
	7	6	5	4	3	2	1	0
bit Symbol	IGRG47	IGRG46	IGRG45	IGRG44	IGRG43	IGRG42	IGRG41	IGRG40
Read/Write	R/W							
After reset	0							
Function	IGBT Timer value: 7-0 bit data							

<IGRG4[15:0]>: Set the IGBT period.

Note1: Setting Value: MTnIGRG2<MTnIGRG3<MTnIGRG4.
Note2: Write to this register by unit of 16 bits or 32 bits. Writing by unit of 8 bits is prohibited.

10.4.1.18 IGBT EMG control register <IGBT mode>

MTnIGEMGCR (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	IGEMGCNT3	IGEMGCNT2	IGEMGCNT1	IGEMGCNT0	—	IGEMGRS	IGEMGOC	IGEMGEN
Read/Write	R/W	R/W	R/W	R/W	R	W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Select the duration of noise elimination for GEMG input 0000: No elimination (Note) 0001: eliminate pulses shorter than 16 / fsys[s] 0010: eliminate pulses shorter than 32 / fsys[s] : 1111: eliminate pulses shorter than 240 / fsys[s]				"0" is read	Cancel EMG state 0: Don't care 1: Cancel	MTOUT0n / MTOUT1n output control in EMG 0: Inactive level 1: Hi-z	EMG circuit control 0: Disable 1: Enable

<IGEMGCNT3:0>: Select the duration of noise elimination for GEMGn input
The noise elimination time is calculated in the formula shown in below.
IGEMGCNT[3:0]×16 / fsys

0000: No elimination
0001: eliminate pulses shorter than 16 / fsys[s]
0010: eliminate pulses shorter than 32 / fsys[s]
0011: eliminate pulses shorter than 48 / fsys[s]
0100: eliminate pulses shorter than 64 / fsys[s]
0101: eliminate pulses shorter than 80 / fsys[s]
0110: eliminate pulses shorter than 96 / fsys[s]
0111: eliminate pulses shorter than 112 / fsys[s]
1000: eliminate pulses shorter than 128 / fsys[s]
1001: eliminate pulses shorter than 144 / fsys[s]
1010: eliminate pulses shorter than 160 / fsys[s]
1011: eliminate pulses shorter than 176 / fsys[s]
1100: eliminate pulses shorter than 192 / fsys[s]
1101: eliminate pulses shorter than 208 / fsys[s]
1110: eliminate pulses shorter than 224 / fsys[s]
1111: eliminate pulses shorter than 240 / fsys[s]

<IGEMGRS>: Cancel the emergency output state. Upon canceling the state, this bit is automatically cleared to “0”
“0”: don’t care
“1”: Cancel

<IGEMGOC>: MTOUT0n/1n output control in EMG
“0”: Inactive level
“1”: Hi-z

<IGEMGEN>: EMG circuit control
“0”: Disable
“1”: Enable

10.4.1.19 IGBT EMG status register <IGBT mode>

MTnIGEMGST (n=0, 1, 2)

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read							
	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	IGEMGIN	IGEMGST
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	"0" is read						Status of EMG input after noise elimination 0: Low 1: High	Status of EMG protection 0: Normal operation 1: Protected

<IGEMGIN>: Status of EMG input after noise elimination
"0": Low
"1": High

<IGEMGST>: Status of EMG protection
"0": Normal operation
"1": Protect
The EMG protection state can be known by reading this bit.

10.4.1.20 Registers for PMD function <PMD mode>

Please refer to PMD mode (Programmable Motor Driver).

10.5 Description of Operations for Timer Mode

The channels operate in the same way, except for the differences in their specifications as shown in Table 10-1. Therefore, the operational descriptions here are only for channel 0.

10.5.1 Configuration

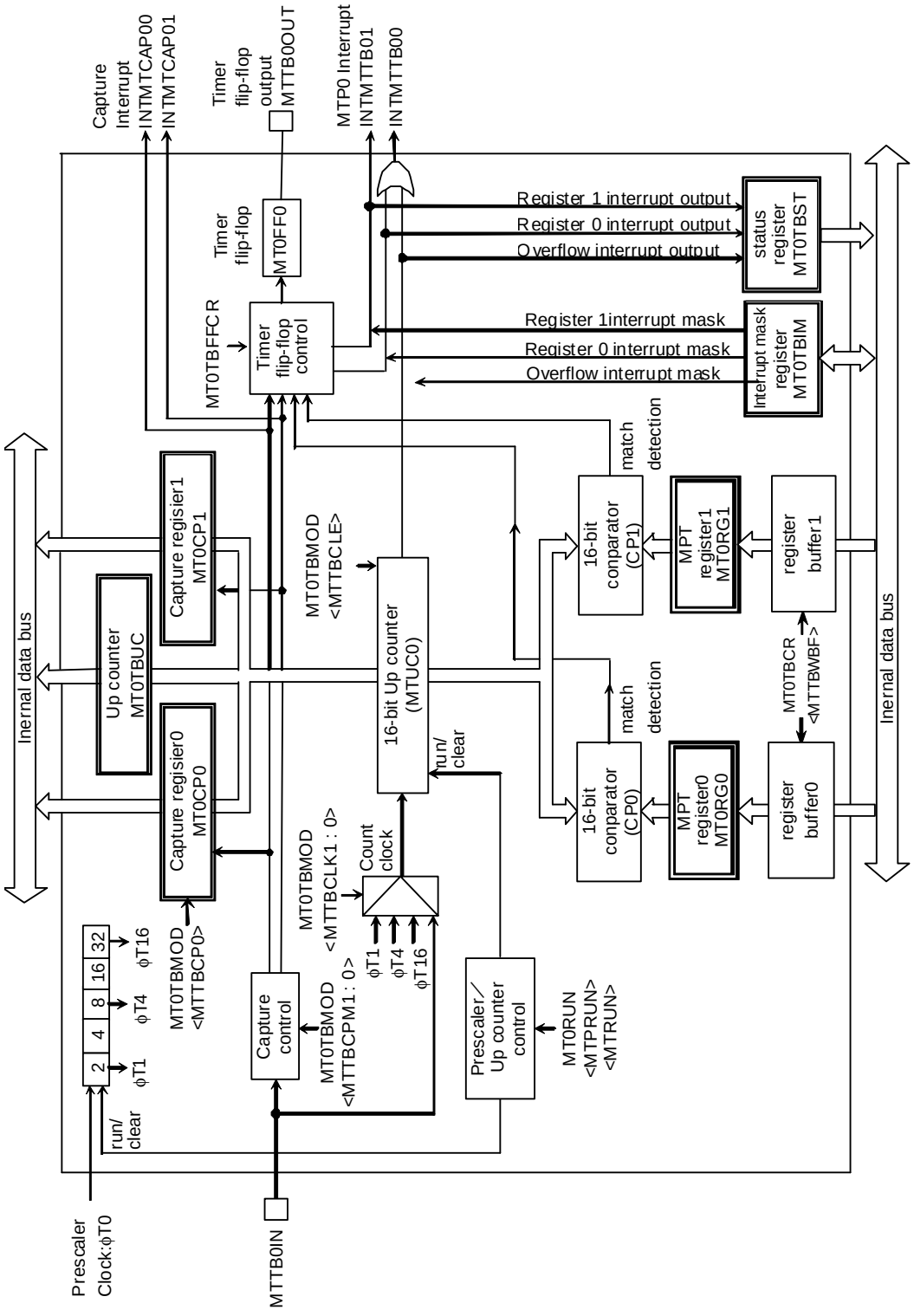


Figure 10-2 Timer Mode Block Diagram

10.5.2 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter MTUC0. The prescaler input clock ϕT0 is fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCLK2:0> 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 stop of a prescaler is set with MTP0RUN<MTPRUN> where writing “1” starts counting and writing “0” clears and stops counting. Table 10-5 shows prescaler output clock resolutions.

Table 10-5 Prescaler Output Clock Resolutions at fc = 40MHz

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

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

10.5.3 Up-counter (MTUC0)

MTUC0 is a 16-bit binary counter.

Source clock

MTUC0 source clock, specified by MT0TBMOD<MTTBCLK1:0>, can be selected from either three types ϕT1, ϕT4 and ϕT16 of prescaler output clock or the external clock of the MTTB0IN pin.

Count start/ stop

Counter operation is specified by MT0RUN<MTRUN>. MTUC0 starts counting if <MTRUN> = “1”, and stops counting and clears counter value if <MTRUN> = “0”.

Timing to clear MTUC0

1) When a compare match is detected

By setting MT0TBMOD<MTTBCLE> = “1”, MTUC0 is cleared in case the comparator detects a match between counter value and the value set in MT0RG1. MTUC0 operates as a free-running counter if MT0TBMOD<MTTBCLE> = “0”.

2) When MTUC0 stops

MTUC0 stops counting and clears counter value if MT0RUN <MTRUN> = “0”.

MTUC0 overflow

If MTUC0 overflow occurs, the INTMTTB00 overflow interrupt is generated.

10.5.4 Timer registers (MT0RG0, MT0RG1)

MT0RG0 and MT0RG1 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 MTUC0 up-counter, it outputs the match detection signal.

Configuration

MT0RG0 and MT0RG1 of this timer registers are paired with register buffer - the double-buffered configuration. The two registers use MT0TBCR<MTTBWBF> to control the enabling/disabling of double buffering. If <MTTBWBF> = “0”, double buffering is disabled and if <MTTBWBF> = “1”, it is enabled. If double buffering is enabled, data is transferred from register buffer to the MT0RG0 and MT0RG1 timer registers when there is a match between MTUC0 and MT0RG1.

Default setting

The values of MT0RG0 and MT0RG1 become undefined after a reset. A reset disables the double buffer.

Register setting

1) When not using double-buffering

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

2) When using double-buffering

MT0RG0/ MT0RG1 and the register buffers are assigned to the same address. If <MTTBWBF> = “0,” the same value is written to MT0RG0, MT0RG1 and each register buffer; if <MTTBWBF> = “1” the value is only written to each register buffer. Therefore, in order to write an initial value to the timer register, the register buffers must be set to “disable”. Then set <MTTBWBF> = “1”and write the following data to the register.

10.5.5 Capture Control

This is a circuit that controls the timing to latch MTUC0 up-counter values into the MT0CP0 and MT0CP1 capture registers. The timing to latch data is specified by MT0TBMOD <MTTBCPM[1:0]>.

Software can also be used to capture values from the MTUC0 up-counter into the capture register; specifically, MTUC0 values are taken into the MT0CP0 capture register each time “0” is written to MT0TBMOD<MTTBCP>. To use this capability, the prescaler must be running (MT0RUN<MTPRUN> = “1”).

10.5.6 Capture Registers (MT0CP0, MT0CP1)

These are 16-bit registers for latching values from the MTUC0 up-counter. To read data from the capture register, use a 16-bit data transfer instruction or read in the order of low-order bits followed by high-order bits.

10.5.7 Up-counter capture register (MT0UC)

Other than the capturing functions shown above, the current count value of the MTUC0 can be captured by reading the MTUC0 registers.

10.5.8 Comparators (MT0CP0, MT0CP1)

These are 16-bit comparators for detecting a match by comparing set values of the MTUC0 up-counter with set values of the MT0RG0 and MT0RG1 timer registers. If a match is detected, INTMTTB00 and INTMTTB01 are generated.

10.5.9 Timer Flip-flop (MT0FF0)

The timer flip-flop (MT0FF0) 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 MT0TBFFCR<MTTBC1T1, MTTBC0T1, MTTBE1T1, MTTBE0T1>.

The value of MT0FF0 becomes undefined after a reset. The flip-flop can be reversed by

writing “00” to MT0TBFFCR<MTTBFF0C[1:0]>. It can be set to “1” by writing “01,” and can be cleared to “0” by writing “10.”

The MT0FF0 value can be output to the timer output pin:MTTB0OUT. To enable timer output, the port C related registers PCCR and PCFR2 must be programmed beforehand.

10.5.10 Capture interrupt (INTMTCAP00, INTMTCAP01)

Interrupts INTMTCAP00 and INTMTCAP01 can be generated at the timing of latching values from the MTUC0 up-counter into the MT0CP0 and MT0CP1 capture registers. The interrupt setting is specified by the CPU.

10.6 Description of Operations for IGBT mode (IGBT)

The channels operate in the same way, except for the differences in their specifications as shown in Table 10-1. Therefore, the operational descriptions here are only for channel 0.

10.6.1 Configuration

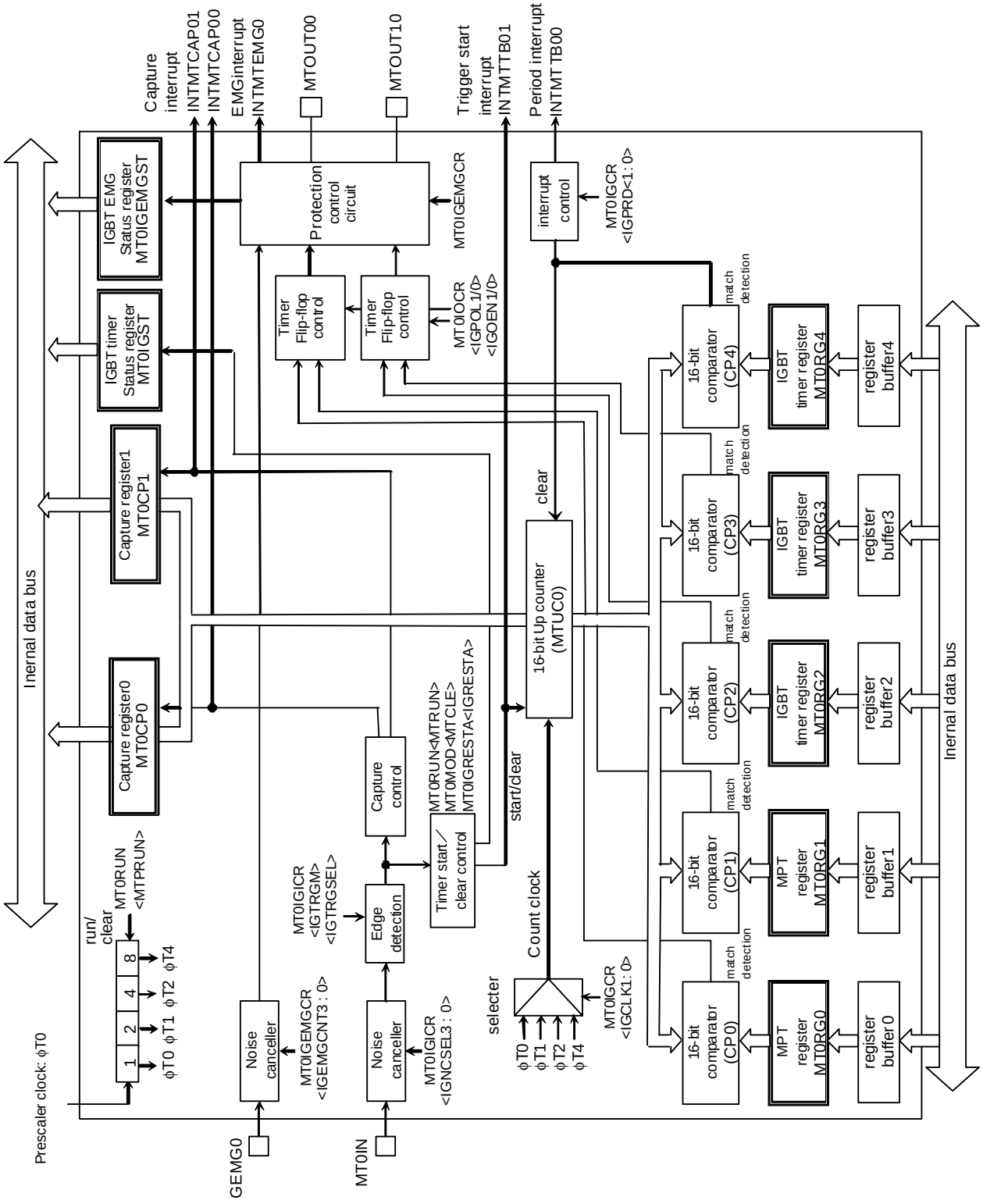


Figure 10-3 IGBT Mode Block Diagram

10.6.1 Prescaler

There is a 4-bit prescaler to generate the source clock for up-counter MTUC0. The prescaler input clock ϕT0 is fperiph, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32 selected by CGSYSCR<PRCLK2:0> 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 stop of a prescaler is set with MT0RUN<MTPRUN> where writing “1” starts counting and writing “0” clears and stops counting. Table 10-6 shows prescaler output clock resolutions.

Table 10-6 Prescaler Output Clock Resolutions at fc = 40MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution			
			φT0	φT1	φT2	φT4
0 (fgear)	000 (fc)	000(fperiph/1)	fc(0.025μs)	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)
		001(fperiph/2)	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)
		010(fperiph/4)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		011(fperiph/8)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		100(fperiph/16)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		101(fperiph/32)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
	100(fc/2)	000(fperiph/1)	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)
		001(fperiph/2)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		010(fperiph/4)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		011(fperiph/8)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		100(fperiph/16)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
		101(fperiph/32)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)	fc/2^9(12.8μs)
	101(fc/4)	000(fperiph/1)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		001(fperiph/2)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		010(fperiph/4)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		011(fperiph/8)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
		100(fperiph/16)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)	fc/2^9(12.8μs)
		101(fperiph/32)	fc/2^7(3.2μs)	fc/2^8(6.4μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)
	110(fc/8)	000(fperiph/1)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		001(fperiph/2)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		010(fperiph/4)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
		011(fperiph/8)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)	fc/2^9(12.8μs)
		100(fperiph/16)	fc/2^7(3.2μs)	fc/2^8(6.4μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)
		101(fperiph/32)	fc/2^8(6.4μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
1 (fc)	000 (fc)	000(fperiph/1)	fc(0.025μs)	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)
		001(fperiph/2)	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)
		010(fperiph/4)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		011(fperiph/8)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		100(fperiph/16)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		101(fperiph/32)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
	100(fc/2)	000(fperiph/1)	–	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)
		001(fperiph/2)	fc/2^1(0.05μs)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)
		010(fperiph/4)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		011(fperiph/8)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		100(fperiph/16)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		101(fperiph/32)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
	101(fc/4)	000(fperiph/1)	–	–	fc/2^2(0.1μs)	fc/2^3(0.2μs)
		001(fperiph/2)	–	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)
		010(fperiph/4)	fc/2^2(0.1μs)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		011(fperiph/8)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		100(fperiph/16)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		101(fperiph/32)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)
	110(fc/8)	000(fperiph/1)	–	–	–	fc/2^3(0.2μs)
		001(fperiph/2)	–	–	fc/2^3(0.2μs)	fc/2^4(0.4μs)
		010(fperiph/4)	–	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)
		011(fperiph/8)	fc/2^3(0.2μs)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)
		100(fperiph/16)	fc/2^4(0.4μs)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)
		101(fperiph/32)	fc/2^5(0.8μs)	fc/2^6(1.6μs)	fc/2^7(3.2μs)	fc/2^8(6.4μs)

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

10.6.3 Up-counter (MTUC0)

MTUC0 is a 16-bit binary counter.

Source clock

MTUC0 source clock, specified by MT0IGCR<IGCLK1:0>, can be selected from either four types ϕT0, ϕT 1, ϕT2 and ϕT4 of prescaler output clock.

Count start/ stop

Counter operation is specified by MT0RUN<MTRUN>. MTUC0 starts counting if <MTRUN> = “1”, and stops counting and clears counter value if <MTRUN> = “0”.

Also, counter is cleared if MT0IGRESTA<IGRESTA> = “1”, and then restarts count-up from 0.

Timing to clear the count

1) When a comparing is matched.

When the comparator detects a match between up-counter value and a value set in MT0IGRG4, counter is cleared.

2) When counter stops

By setting MT0RUN<MTRUN>=”0”, counter stops and is cleared.

3) When counter restart

By setting MT0IGRESTA<IGRESTA>=”1”, counter is cleared and starts counting up from “0”.

4) When trigger start mode is set.

In the trigger start mode, by setting MT0IN pin is driven to the specified stop level, counter stops and is cleared.

Count up and count clear operation

Here is the description of count and clear operation and period setup both when source clock ϕT0 is selected and when source clock ϕT1, ϕT2 or ϕT4 is selected.

1) When source clock ϕT0 is selected

When source clock ϕT0 is selected, two source clocks are required for the match count and the clear count. Thus a value of the period setup is set to M＋1.

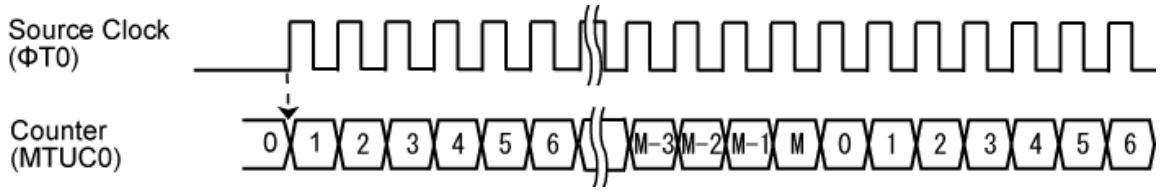


Figure 10-4 Count-up/ count clearing when source clock ϕT0 is selected.

2) Other than source clock φT0 is selected

When source clock φT1, φT2 or φT4 is selected, one source clock is required for the match count and the clear count. Thus a value of period setup is set to M.

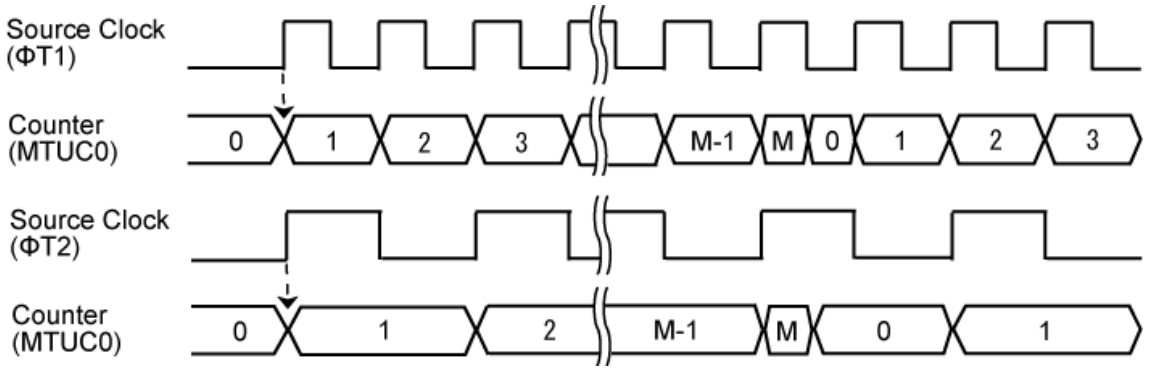


Figure 10-5 Count-up/ count clearing when source clock **ϕ**T1, **ϕ**T2 or **ϕ**T4 is selected.

10.6.4 Period Setup Register (MT0IGRG4)

Period Setup register is a double-buffered configuration register which sets PPG output periods. If the MT0IGR4 matches with up-counter (MTUC0), counter is cleared and data is updated on next period. At this timing, data is transferred to timer register (MT0IGRG4) from register buffer 4.

10.6.5 Timer Registers (MT0RG0, MT0RG1, MT0IGRG2, MT0IGRG3)

Timer registers are registers for setting values to compare with up-counter values. If the comparator detects a match between a values set in these timer registers, it outputs a match detection signal. Timer registers (MT0RG0, MT0RG1, MT0IGRG2, MT0IGRG3) are the double-buffered configuration and are paired of register buffer. If MT0IGRG4 matches with up-counter (MTUC0), counter is cleared at the same time data is updated. At this time, data is transferred to timer register (MT0IGRG2 or 3) from register buffer 2 or 3.

In the IGBT mode, MT0RG0 and 1 are always doubled-buffered configuration.

Write and read operation of timer register (MT0RG0, MT0RG1, MT0IGR2, MT0IGRG3) and period setup register (MT0IGRG4)

1) When timer registers and period setup register are written

When timer is stopping, data can be written into timer register (MT0RG0, MT0RG1, MT0IGRG2, MT0IGRG3) and period setup register (MT0IGRG4).

When timer is in operation, data is latched in each register buffer. If MT0IGRG4 matches with up-counter (MTUC0), counter is cleared at the same time data is updated.

2) When timer registers and period setup register are read

Current value of 16-bit comparator and the target register for comparison to be read. A value of register buffer cannot be read.

Note)

Write to this register by unit of 16 bits or 32 bits. Writing in unit of 8 bits is prohibited.

10.6.6 Capture Control

By setting to command start mode or trigger capture mode, up-counter values (MTUC0) are captured by MT0CP0 and MT0CP1 at the rising edge and falling edge of MT0IN pin respectively.

10.6.7 Capture Registers (MT0CAP0, MT0CAP1)

These are registers for latching values from the up-counter (MTUC0).

10.6.8 Comparators (CP0, CP1, CP2, CP3, CP4)

These are comparators for detecting a match by comparing set values of the MTUC0 up-counter with set values of the timer registers (MT0RG0, MT0RG1, MT0IGRG2, MT0IGRG3 and MT0IGRG4) to detect a match.

10.6.9 Output Signal Control (MTOUT00, MTOUT01)

Output Signal Control signals (MTOUT00, MTOUT01) are changing by a match detect signal between up-counter and timer register. The initial mode setting of output control signals are specified with MT0IGOCR<IGPOL0,1>. After reset, these signals are set to “Low” (MT0IGOCR<IGPOL0,1>=0). If MT0IGOCR<IGPOL0,1> is set to “0”, the initial state is “Low”. If MT0IGOCR<IGPOL0,1> is set to “1”, the initial state is “High”.

Output control signals are specified with MT0IGOCR<IGOEN0,1>. After reset, these signals are disabled. When using these signals, set MT0IGOCR<IGOEN0,1>=1.

10.6.10 Capture Interrupt (INTMTCAP00, INTMTCAP01)

Interrupts INTMTCAP00 and INTMTCAP01 can be generated at the timing of latching values from the MTUC0 up-counter into the MT0CP0 and MT0CP1 capture registers. The interrupt setting is specified with a register of CPU.

10.6.11 Trigger Start Interrupt (INTMTTB01)

In the case that the command start and trigger start mode, or the trigger start mode is selected, a trigger start Interrupt occurs when the counter starts upon the detection of a trigger edge specified with MTnIGCR<IGTRGSEL>. In the trigger capture mode, trigger edge detection does not cause an interrupt. A start trigger causes an interrupt even when the counter is stopped in emergency.

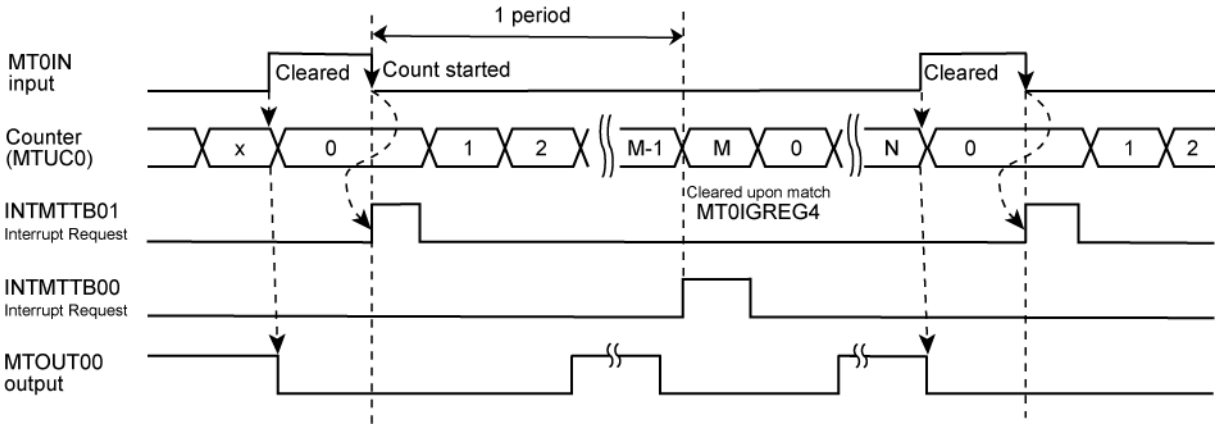


Figure 10-6 Trigger Start Interrupt Operation

10.6.12 Period Setup Interrupt (INTMTTB00)

In the case that command start and trigger capture mode, or command start and trigger start mode is selected, a period interrupt (INTMTTB00) occurs when the counter starts with a command start and when the counter is cleared with the specified counter period (MTnIGREG4) reached at the end of specified period. A match with the set period causes an interrupt even when the counter is stopped in emergency. The interrupt periods are selected as every 1-period, 2-period and 4-period specified with MTnIGCR<IGPRD[1:0]>.

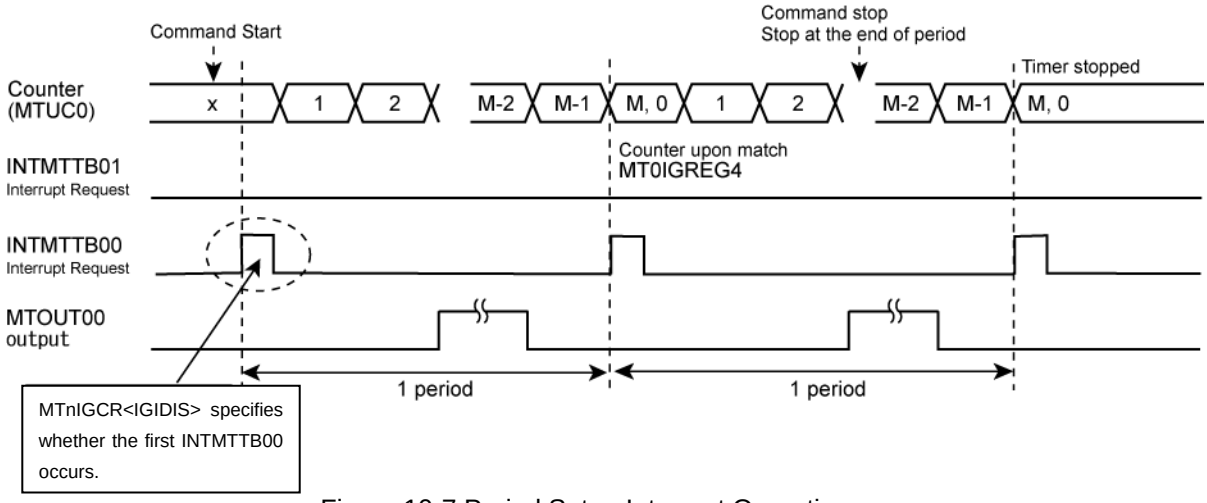


Figure 10-7 Period Setup Interrupt Operation

An interrupt generation at the start counting is prohibited or permitted with command start interrupt control register MT0IGCR<IGIDIS>. When a command starts specified with MT0RN<MTRUN> setting to 1" and MT0IN pin is stop level, the counter does not start (INTMTTB00 is not generated). A subsequent trigger start edge causes the counter to start and INTMTTB01 occurs.

10.6.13 Basic operation

To start timer after setting MT0RG0, 1 and MT0IGRG2 to MT0IGRG4, PPG pulses are outputted to MTOUT00 pin and MTOUT10 pin.

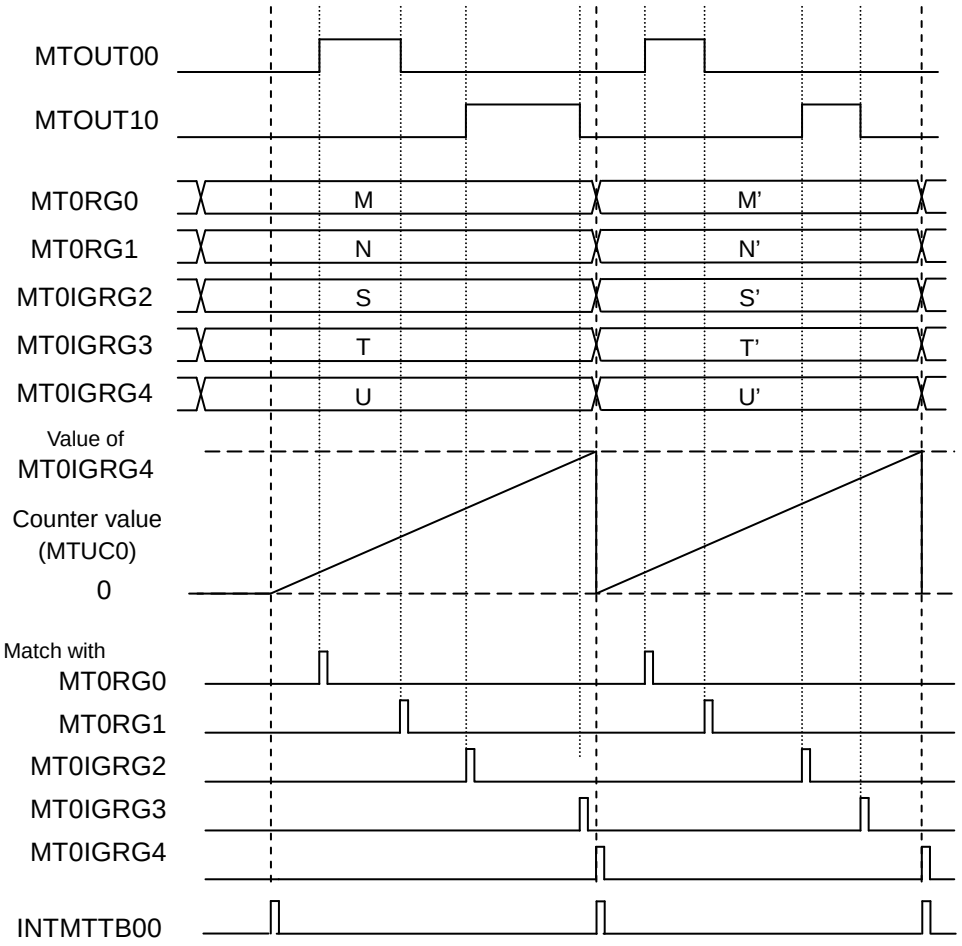


Figure 10-8 Basic timing of IGBT function

10.6.14 How to start IGBT timer

IGBT modes are selected among three starting modes.

10.6.14.1 Command start and Trigger capture mode

Writing “1” to MT0RUN<MTRUN> causes the current count to be cleared and the counter to start counting. Once the count has reached a specified period, the counter is cleared. The counter subsequently restarts counting if specifies continuous mode (MT0IGCR<IGSNGL>=0) it stops counting if specifies one-time mode (MT0IGCR<IGSNGL>=1). Writing “1” to MT0IGRESTA <IGRESTA> before the count reaches a period causes the counter to be cleared, after which it operates as specified with <IGSTP[1:0]>.

The count values at the rising and falling edges on the MT0IN pin can be stored in capture registers.

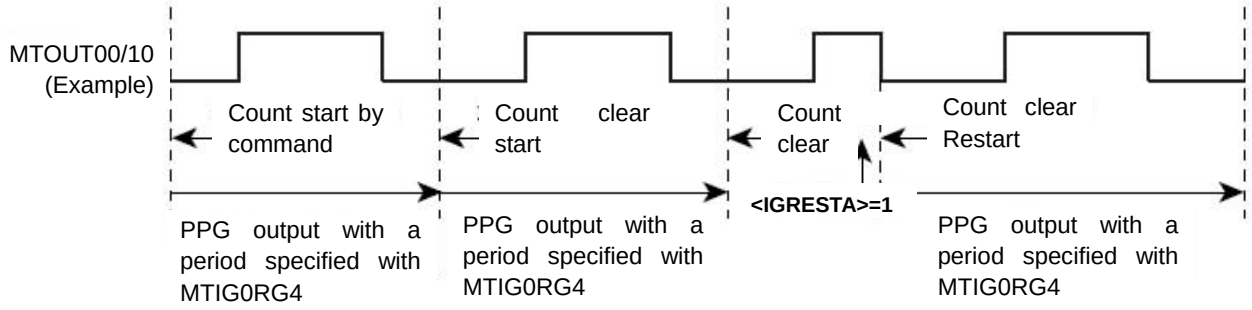


Figure 10-9 Operation in Command start

When the counter starts in the command start and trigger capture mode, counter values of MTnIN input both at the rising edge and falling edge are captured by the MTnCPA0 and MTnCAP1 capture register respectively. When the values are captured, INTMTCAPn0 and INTMTCAPn1 generate respectively.

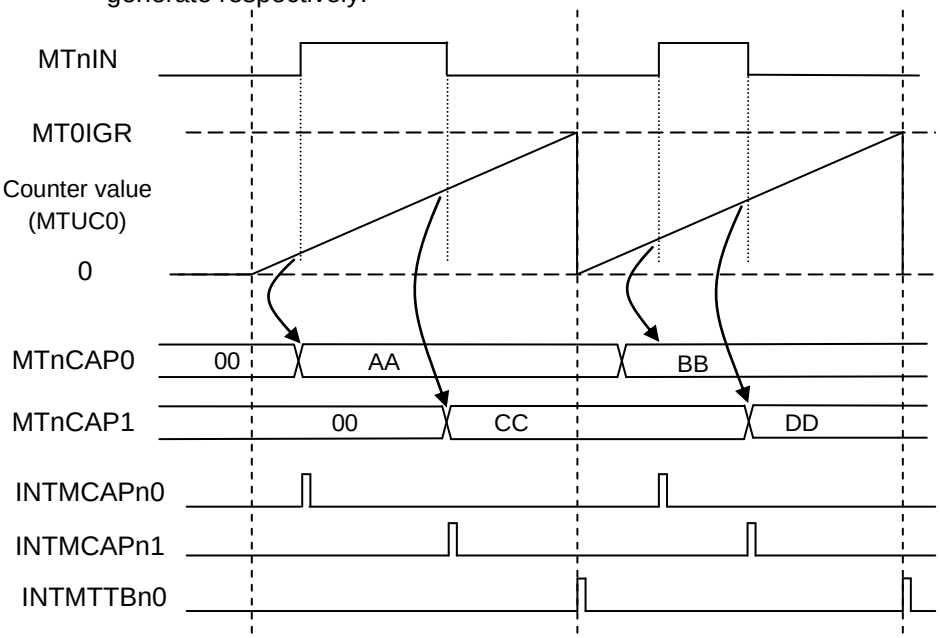


Figure 10-10 Operation in Capture

10.6.14.2 Command start and Trigger start mode

Writing a 1 to MT0RUN<MTRUN> causes the current count to be cleared and the counter to start counting. The operation is the same as that in command start and capture mode if there is no trigger input on the MT0IN pin. If an edge specified with the start edge selection field (MT0IGICR <IGTRGSEL>) appears on the MT0IN pin, however, the timer starts counting. The counter is cleared and stopped while the MT0IN pin is driven to the specified clear/stop level. If the MT0IN pin is at the clear/stop level when a count start command is issued (1 is written to MT0RUN <MTRUN>), counting does not start (INTMTTB00 does not occur) until a trigger start edge appears, causing INTMTTB01 to occur (A trigger input takes precedence over a command start).

Note: For more information on the acceptance of a trigger, see “Trigger start/stop acceptance mode”.

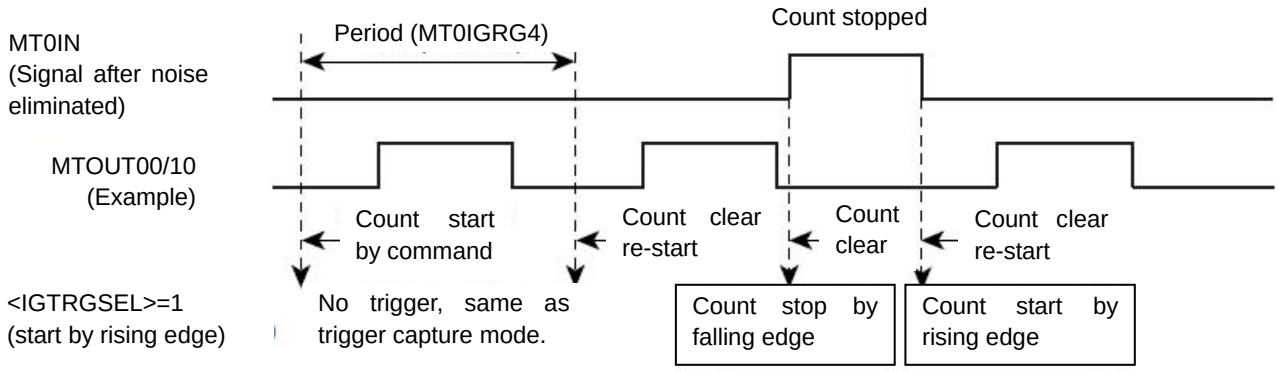


Figure 10-11 Operation in Command start and Trigger start mode

10.6.14.3 Trigger start mode

If an edge specified with the start edge selection field (MT0IGICR<IGTRGSEL>) appears on the MT0IN pin, the timer starts counting. The counter is cleared and stopped while the MT0IN pin is driven to the specified clear/stop level.

In trigger start mode, writing a 1 to MT0RUN<MTRUN> is ignored and does not initialize the MPT0OUT0/1 outputs.

Note: For more information on the acceptance of a trigger, see “Trigger start/stop acceptance mode”.

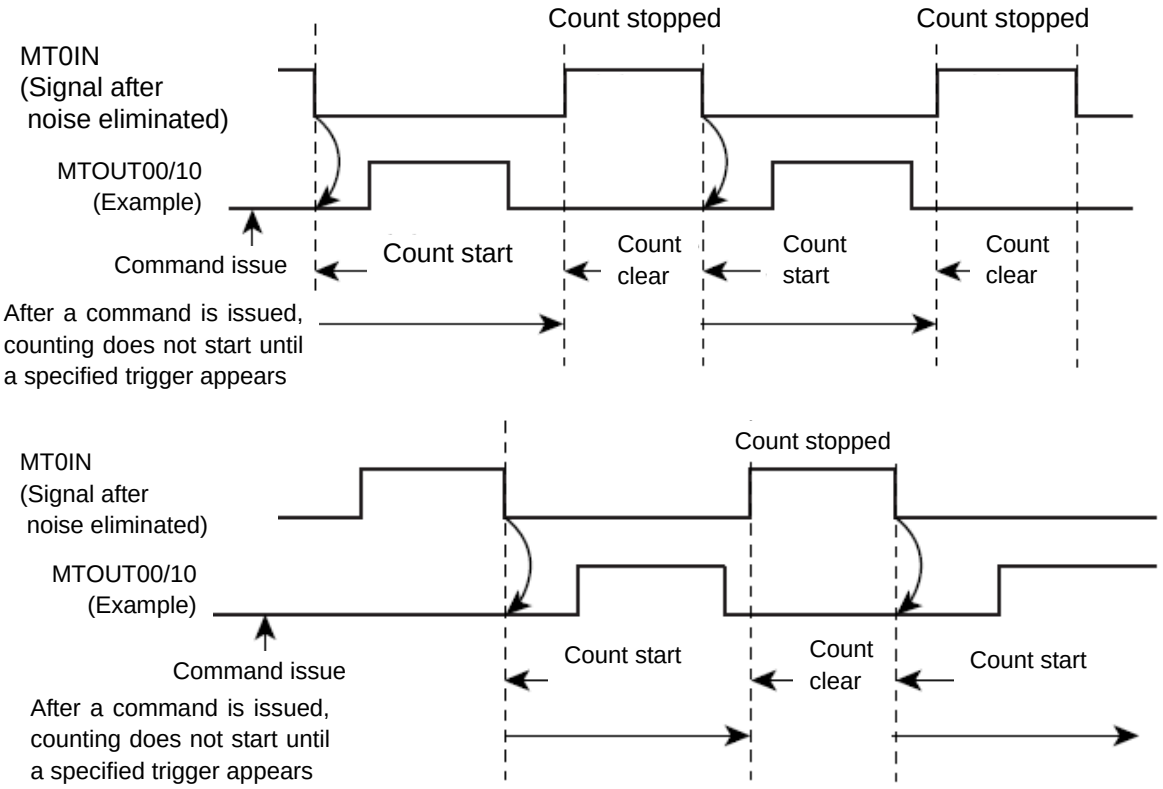


Figure 10-12 Operation in Trigger start mode

10.6.15 One-time / continuous output mode

The IGBT output can be set to either one-time output mode or continuous output mode.

10.6.15.1 Continuous output mode

Setting the timer (MT0IGCR<IGSNGL> = “0”) when the timer starts (MT0RUN<MTRUN> = “1”) specifies the continuous output mode. In this mode, the timer continuously outputs specified waveforms.

10.6.15.2 One-time output mode

Setting the timer (MT0IGCR<IGSNGL> =”1”) when the timer starts (MT0RUN<MTRUN> = “1”) specifies the one-time output mode. In this mode, the timer stops the counter at the end of a single period.

For a trigger start, the counter stops until a trigger is detected. A specified trigger starts counting and the counter stops at the end of a single period. Set “1” to MTnRUN<MTRUN> to restart the counter using a trigger.

10.6.16 Configuring how the timer stops

Setting MT0RUN<MTRUN> to 0 causes the timer to stop with the specified output state according to the setting of MT0IGOCR<IGSTP>.

10.6.16.1 Counting stopped with outputs initialized

When MT0IGOCR<IGSTP> is set to “00”, the counter stops immediately with the MTOUT00 and MTOUT10 outputs initialized values specified with MT0IGOCR<IGPOL[1:0]>.

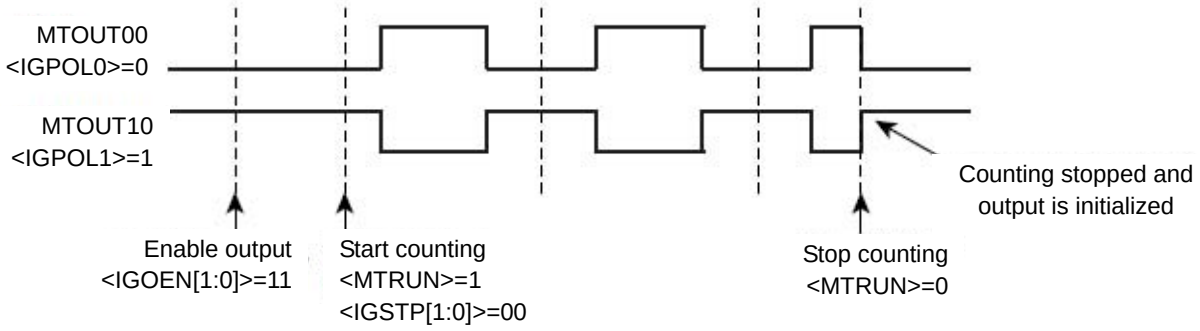


Figure 10-13 Immediately Stopping and Clearing the Counter with the Outputs Initialized (<IGSTP>=00)

10.6.16.2 Counting stopped with outputs maintained

When MT0IGOCR<IGSTP> is set to “01”, the counter stops immediately with the current MTOUT00 and MTOUT10 output states maintained.

To restart the counter from the maintained state (MT0IGOCR<IGSTP>=01), set MT0RUN <MTRUN> to “1”. The counter is restarted with the initial output values, specified with MT0IGOCR<IGPOL[1:0]>.

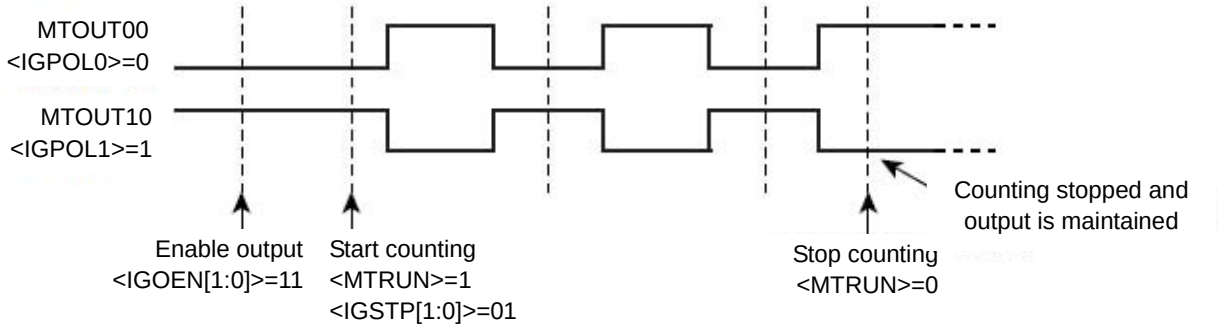


Figure 10-14 Immediately Stopping and Clearing the Counter with the Outputs maintained (<IGSTP>=01)

10.6.16.3 Counting stopped with outputs initialized at the end of the period

When MT0IGOCR<IGSTP> is set to “10”, the counter continues counting until the end of the current period and then stops. If a stop trigger is detected before the end of the period, however, the counter stops immediately.

Registers except MT0IGOCR must not be rewritten before the counter stops completely.

MT0IGST<IGST> can be read to determine whether the counter has stopped.

Note that before changing the timer settings, confirm the counter stopped after period completed.

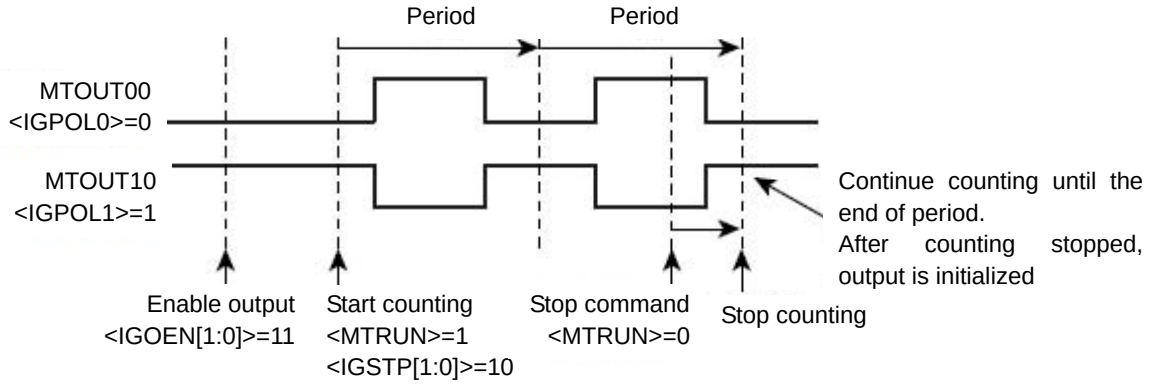


Figure 10-15 Stopping the Counter at the End of the Period (<IGSTP>=10)

10.6.17 Trigger input

10.6.17.1 Selecting an input signal logic

The logic for an input trigger signal on the MT0IN pin can be specified using MT0IGICR<IGTRGSEL>.

- IGTRGSEL = “0”:
  Counting starts on the rising edge. The counter is cleared and stopped while the MT0IN pin is low.
- IGTRGSEL =” 1”:
  Counting starts on the falling edge. The counter is cleared and stopped while the MT0IN pin is high.

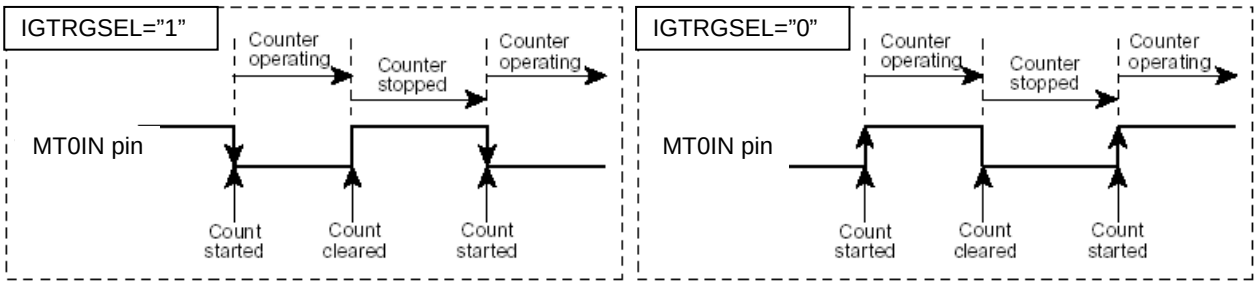


Figure 10-16 Trigger input Signal

In one-time stop mode, the counter accepts a stop trigger but does not accept a start trigger (when a stop trigger is accepted within a period, the output is immediately initialized and the counter is stopped).

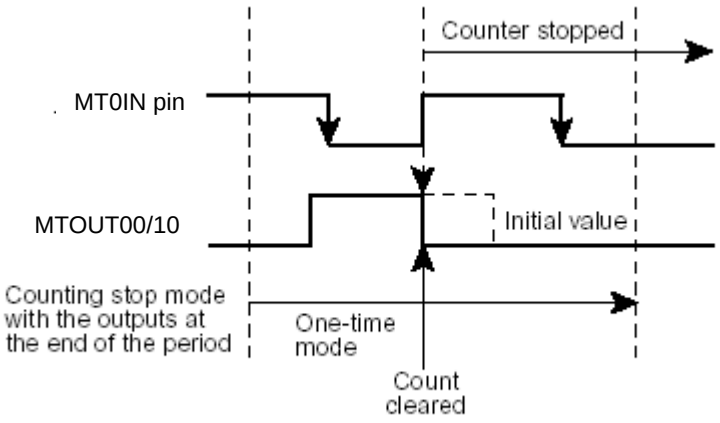


Figure 10-17 Trigger accepted in Stop counter after completing output in the current period

All triggers (Start and stop) are ignored when the timer is stopped (MT0RUN <MTRUN> = “0”).

10.6.17.2 Specifying whether trigger are always accepted or ignored when MTOUT00/10 outputs are active

The MT0IGICR<IGTRGM> specifies whether triggers from the MT0IN pin are always accepted or ignored when the PPG output is active.

• IGTRGM = “0”:

Triggers from the MT0IN pin are always accepted regardless of whether MTOUT00 and MTOUT10 outputs are active or inactive. A trigger starts or clears/stops the timer and deactivates MTOUT00 and MTOUT10 outputs.

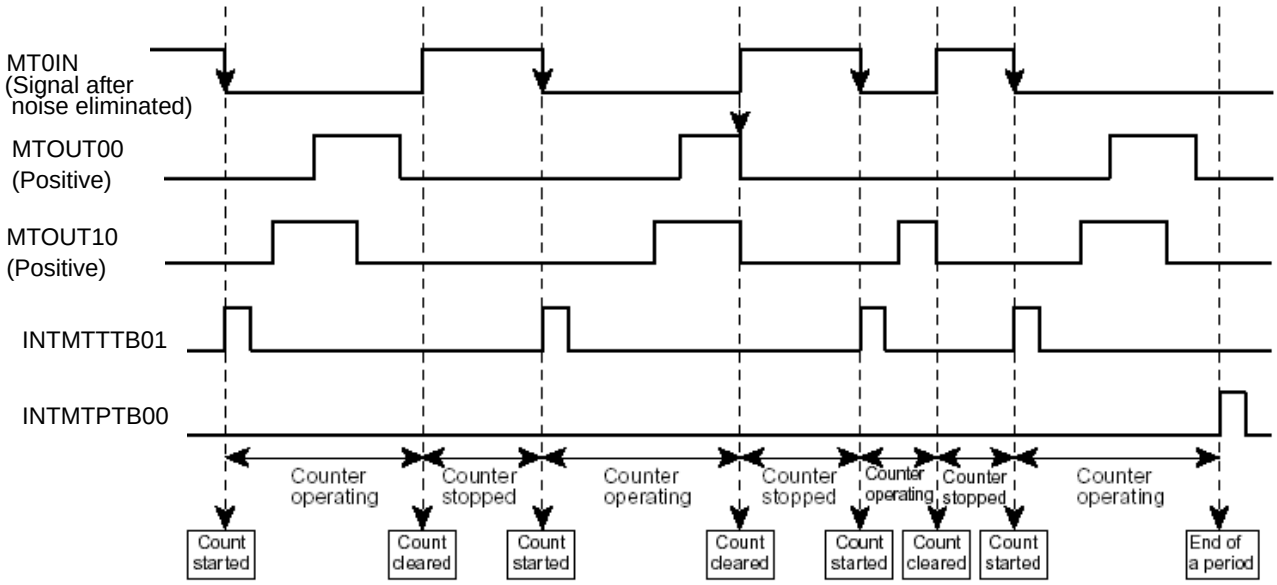


Figure 10-18 Start Trigger on the MT0IN pin

• IGTRGM = “1”:

Triggers from the MT0IN pin are accepted only when MTOUT00 and MTOUT10 outputs are inactive. A trigger starts or clears/stops the timer. Triggers are ignored when MTOUT00 and MTOUT10 outputs are active.

The active/inactive state of the MTOUT00 and MTOUT10 pins has meaning only when output on the pin is enabled with <IGOEN0> or <IGOEN1>.

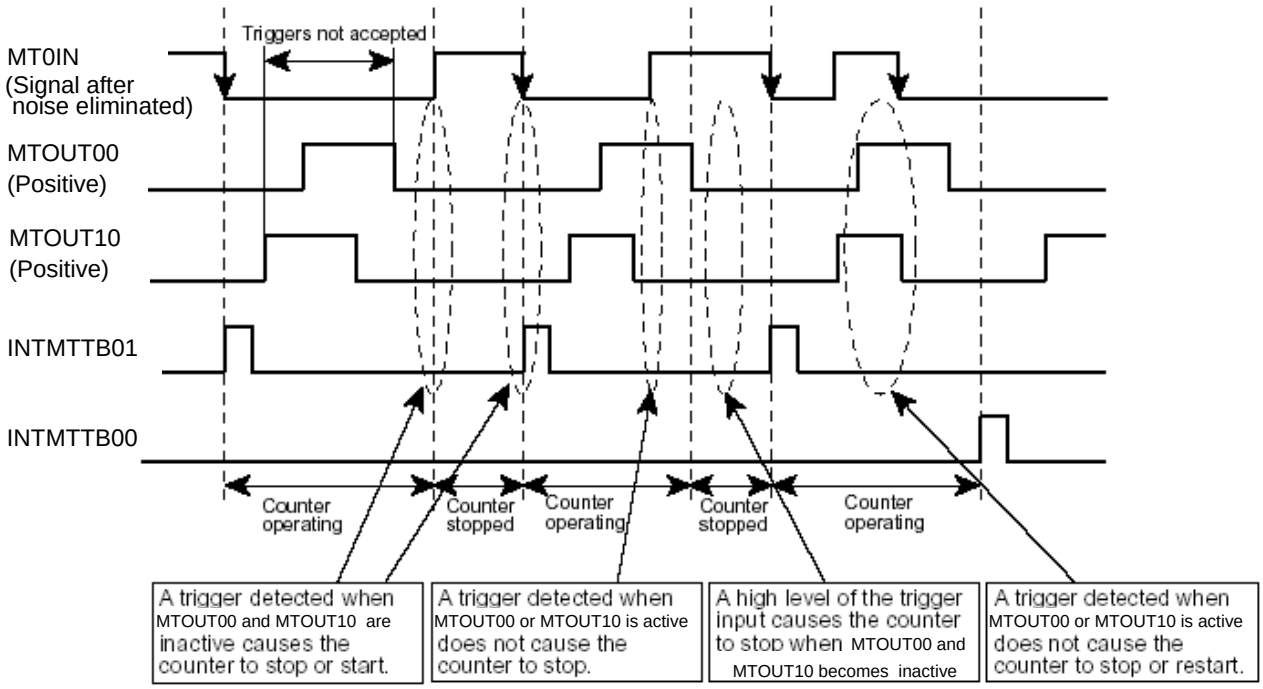


Figure 10-19 Trigger start

10.6.18 Noise canceller

A signal from external input pin (MTnIN, GEMGn) is eliminated noise by digital noise canceller. Digital noise canceller can be selected noise eliminating time with MTnIGIGR<IGNCSEL3:0> and MTNIGEMGCR<IGEMGCNT3:0>.

10.6.19 Emergency stop function

10.6.19.1 Operating Description

When MTnIGEMGCR<IGEMGEN> is set to “1”, the emergency stop function is permitted, that is, GEMGn input is permitted. When a low level input is detected with GEMGn pin, MTOUT0n and MTOUT1n waveform are initialized or become Hi-z state, and the GEMGn interrupt occurs specified with MTnGEMGCR<IGEMGOC> setting.

Note that this function is only prohibited the MTOUT0n and MTOUT1n outputs not the counter stopping thus timer must be stopped in the interrupt routine.

10.6.19.2 Emergency Stop Monitoring

When the emergency output stop feature activates, the MTnIGEMGST<IGEMGST> is set to “1”. If “0” is read from IGEMGST, it indicates that the emergency output stop is enabled.

10.6.19.3 GEMG Interrupt

A GEMG interrupt (INTMTENGn) occurs when an emergency output stop input is accepted. To use a GEMG interrupt (INTMTENGn) for some processing, ensure that the interrupt is permitted beforehand. When GEMGn pin is low, an attempt to cancel the emergency output stop state results in an interrupt being generated again, with the emergency output stop state reestablished.

10.6.19.4 Cancelling the Emergency output stop state

To cancel the emergency output stop state, perform the following steps:

1. Confirm GEMGn pin is high
2. Set MTnRUN<MTRUN> to “0”.
3. Confirm the timer operation stopped (MTnIGST<IGST>=0)

After confirming above steps, by setting to MTnIGEMGCR<IGEMGRS> to “1”, the emergency output stop state is released.

10.7 Description of Operations for PMD mode (PMD: Programmable Motor Driver)

The MPT has programmable motor driver (PMD) module. The PMD supports interaction with the AD converter.

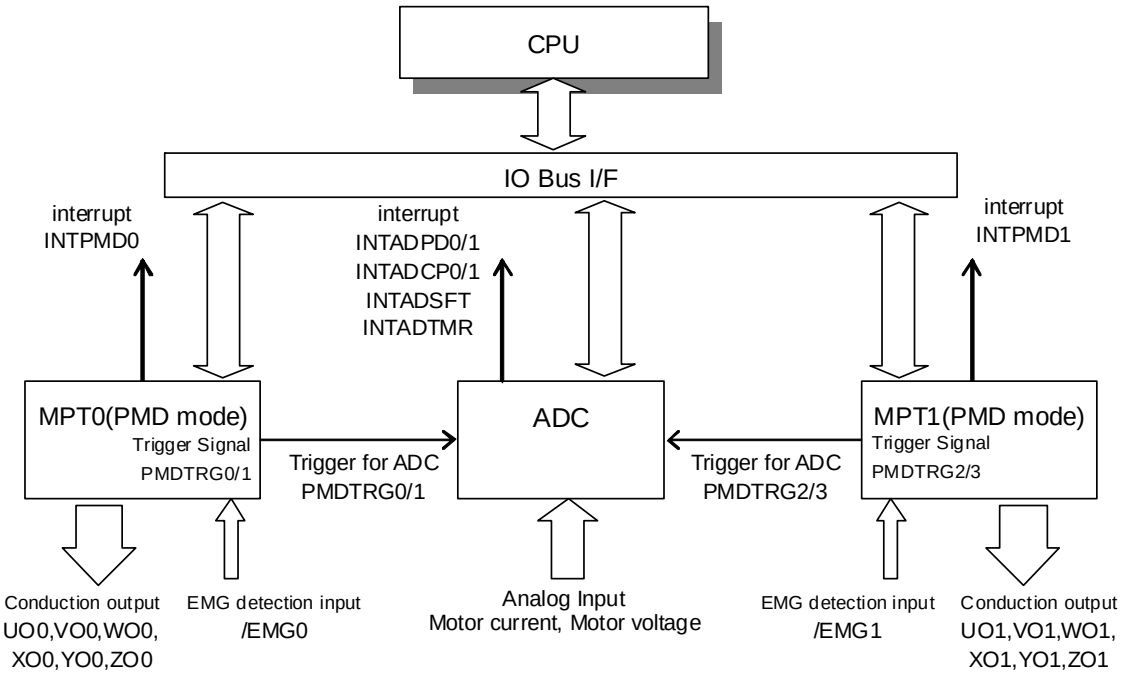


Fig 10-1 Motor Control-related Block Constitution

10.7.1 PMD Input/ Output Signals

The Table 10-4 shows the signals that are input to and output from each PMD (MPT).

Table 10-7 Input/ Output Signals

Channel	Pin Name	PMD Signal Name	Description
CH 0	PC6/$\overline{\text{EMG0}}$	EMG0	EMG state signal
	PC0/UO0	U0	U-phase output
	PC1/XO0	X0	X-phase output
	PC2/VO0	V0	V-phase output
	PC3/YO0	Y0	Y- phase output
	PC4/WO0	W0	W-phase output
	PC5/ZO0	Z0	Z-phase output
CH 1	PG6/$\overline{\text{EMG1}}$	EMG1	EMG state signal
	PG0/UO1	U1	U-phase output
	PG1/XO1	X1	X-phase output
	PG2/VO1	V1	V-phase output
	PG3/YO1	Y1	Y-phase output
	PG4/WO1	W1	W-phase output
	PG5/ZO1	Z1	Z-phase output

10.7.2 PMD registers

Table 10-8 shows a list of PMD register. (Upper case: PMD0, lower case: PMD1)

Table 10-1 List of PMD register

address	Register symbols	Register names
0x4005 0400 0x4005 0480	MTPD0MDEN MTPD1MDEN	PMD Enable Register
0x4005 0404 0x4005 0484	MTPD0PORTMD MTPD1PORTMD	Port Output Mode Control Register
0x4005 0408 0x4005 0488	MTPD0MDCR MTPD1MDCR	PMD Control Register
0x4005 040C 0x4005 048C	MTPD0CNTSTA MTPD1CNTSTA	PWM Counter Status Register
0x4005 0410 0x4005 0490	MTPD0MDCNT MTPD1MDCNT	PWM Counter Register
0x4005 0414 0x4005 0494	MTPD0MDPRD MTPD1MDPRD	PWM Period Register
0x4005 0418 0x4005 0498	MTPD0CMPU MTPD1CMPU	PMD Compare U Register
0x4005 041C 0x4005 049C	MTPD0CMPV MTPD1CMPV	PMD Compare V Register
0x4005 0420 0x4005 04A0	MTPD0CMPW MTPD1CMPW	PMD Compare W Register
0x4005 0424 0x4005 04A4	Reserved	―
0x4005 0428 0x4005 04A8	MTPD0MDOUT MTPD1MDOUT	PMD Output Control Register
0x4005 042C 0x4005 04AC	MTPD0MDPOT MTPD1MDPOT	PMD Output Setting Register
0x4005 0430 0x4005 04B0	MTPD0EMGREL MTPD1EMGREL	EMG Release Register
0x4005 0434 0x4005 04B4	MTPD0EMGCR MTPD1EMGCR	EMG Control Register
0x4005 0438 0x4005 04B8	MTPD0EMGSTA MTPD1EMGSTA	EMG Status Register
0x4005 043C 0x4005 04BC	Reserved	-
0x4005 0440 0x4005 04C0	Reserved	-
0x4005 0444 0x4005 04C4	MTPD0DTR MTPD1DTR	Dead Time Register
0x4005 0448 0x4005 04C8	MTPD0TRGCMP0 MTPD1TRGCMP0	Trigger Compare 0 Register
0x4005 044C 0x4005 04CC	MTPD0TRGCMP1 MTPD1TRGCMP1	Trigger Compare 1 Register
0x4005 0450 0x4005 04D0	Reserved	-
0x4005 0454 0x4005 04D4	Reserved	-
0x4005 0458 0x4005 04D8	MTPD0TRGCR MTPD1TRGCR	Trigger Control Register
0x4005 045C 0x4005 04DC	MTPD0TRGMD MTPD1TRGMD	Trigger Output Mode Setting Register
0x4005 0460 0x4005 04E0	Reserved	-
0x4005 047C 0x4005 04FC	Reserved	-

10.7.3 PMD Module

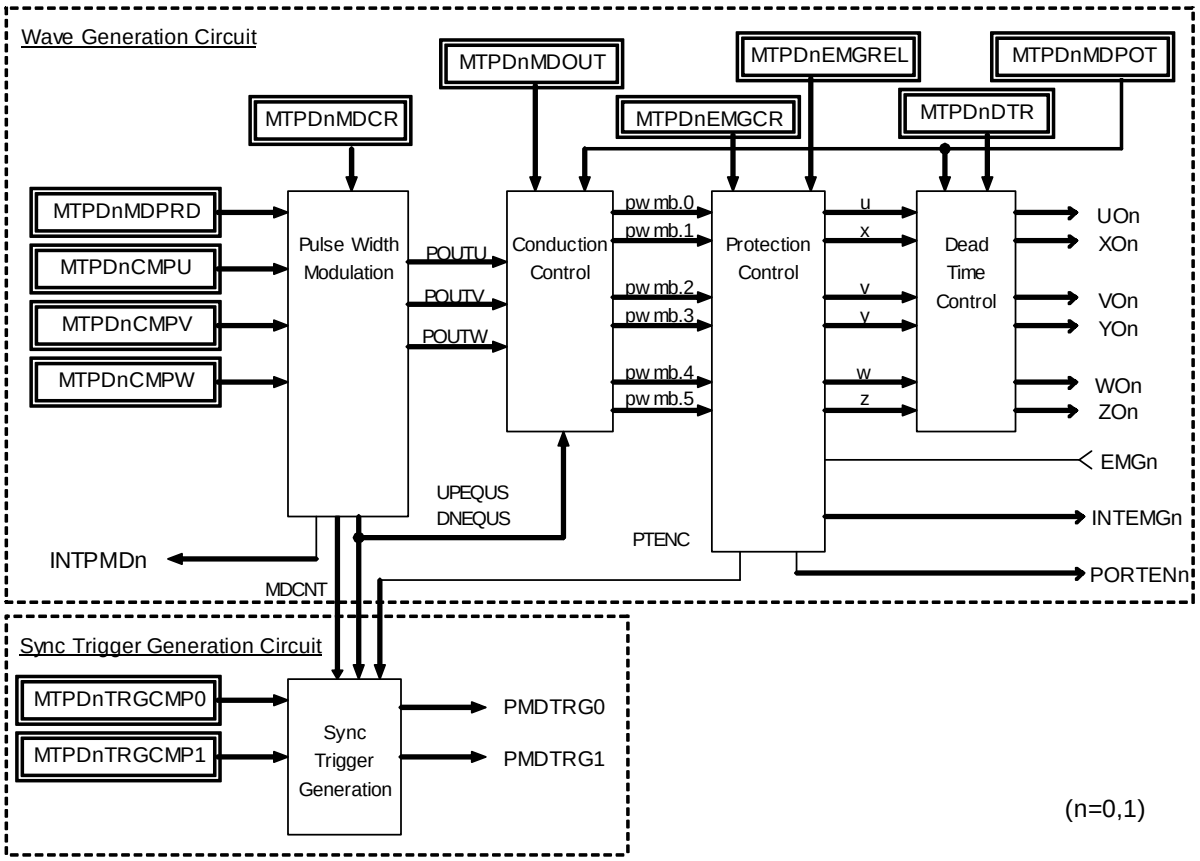


Fig 10-2 Block diagram of PMD Module

The PMD Module consists of two blocks of a wave generation circuit and a sync trigger generation circuit. The wave generation circuit includes a pulse width modulation circuit, a conduction control circuit, a protection control circuit, a dead time control circuit.

- The pulse width modulation circuit generates independent 3-phase PWM waveforms with the same PWM frequency.
- The conduction control circuit determines the output pattern for each of the upper and lower sides of the U, V and W phases.
- The protection control circuit controls emergency output stop by EMG input.
- 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.

- PMD Enable Register (MTPDnMDEN)

0x4005 0400, 0x4005 0480

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	PWMEN
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R/W
After reset	0	0	0	0	0	0	0	0

<PWMEN>: Enables or disables waveform synthesis.
0: Disable
1: Enable

Output ports that are used for the PMD become high impedance when the PMD is disabled. Before enabling the PMD, configure other relevant settings, such as output port polarity.

- Port Output Mode Register (MTPDnPORTMD)

0x4005 0404, 0x4005 0484

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	PORTMD
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R/W	R/W
After reset	0	0	0	0	0	0	0	0

<PORTMD>: Port control setting
0: Upper phases = High-Z
1: Upper phases = PMD output

The <PORTMD> setting controls external port outputs of the upper phases (U, V and W phases) and the lower phases (X, Y and Z phases). When a tool break occurs while “High-Z” is selected, the upper and lower phases of external output ports are set to high impedance. In other cases, external port outputs depend on PMD outputs.

* When PWMEN=0, output ports are set to high impedance regardless of the output port setting.
* When an EMG input occurs, external port outputs are controlled depending on the MTPDnEMGMD setting.

<Bit 1> : Must be write “0”

10.7.3.1 Pulse Width Modulation Circuit

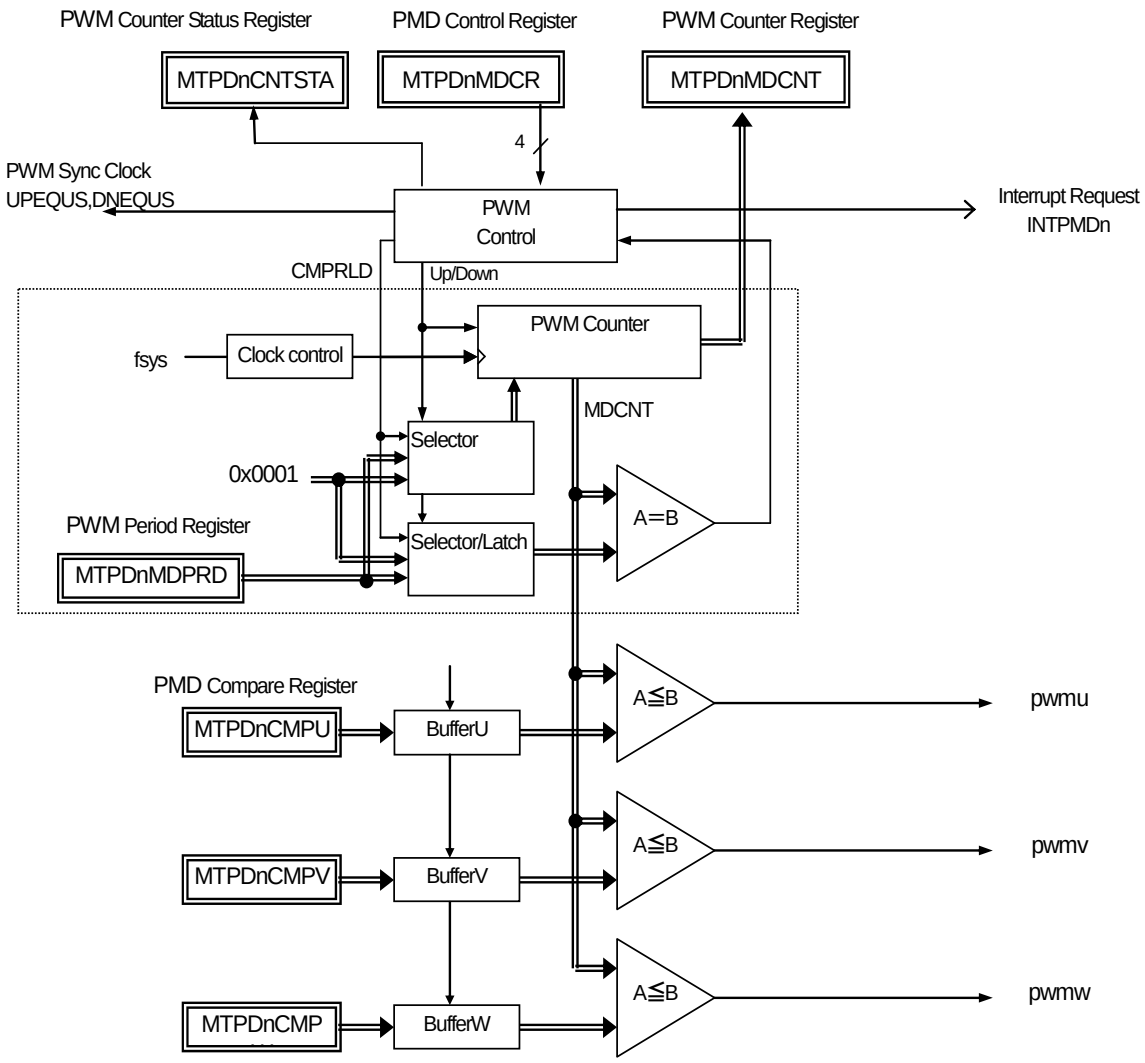


Fig 10-3 Pulse Width Modulation Circuit

The pulse width modulation circuit has a 16-bit PMD up-/down-counter and generates PWM carrier waveforms with a resolution of 25 ns at 40 MHz. The PWM carrier waveform mode can be selected from mode 0 (edge-aligned PWM, sawtooth wave modulation) and mode 1 (center-aligned PWM, triangular wave modulation).

The PWM period extension mode (MTPDnMDCR<PWMCK> = 1) is also available. When this mode is selected, the PWM counter generates PWM carrier waveforms with a resolution of 100 ns.

(1) Setting the PWM period

The PWM period is determined by the MTPDnMDPRD register. This register is double-buffered. Comparator input is updated at every PWM period. It is also possible to update comparator input at every half PWM period.

$$\text{Sawtooth wave PWM: MTPDnMDPRD register value} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]}}$$

$$\text{Triangular wave PWM: MTPDnMDPRD register value} = \frac{\text{Oscillation frequency [Hz]}}{\text{PWM frequency [Hz]} \times 2}$$

(2) Compare function

The pulse width modulation circuit compares the PWM compare registers of the 3 phases (MTPDnCMPU, MTPDnCMPV, MTPDnCMPW) and the carrier wave generated by the PWM counter (MTPDnMDCNT) to determine which is larger to generate PWM waveforms with the desired duty.

The PWM compare register of each phase has a double-buffered compare register. The PWM compare register value is loaded at every PWM period (when the internal counter value matches the MTPDnMDPRD value). It is also possible to update the compare register at every 0.5 PWM periods.

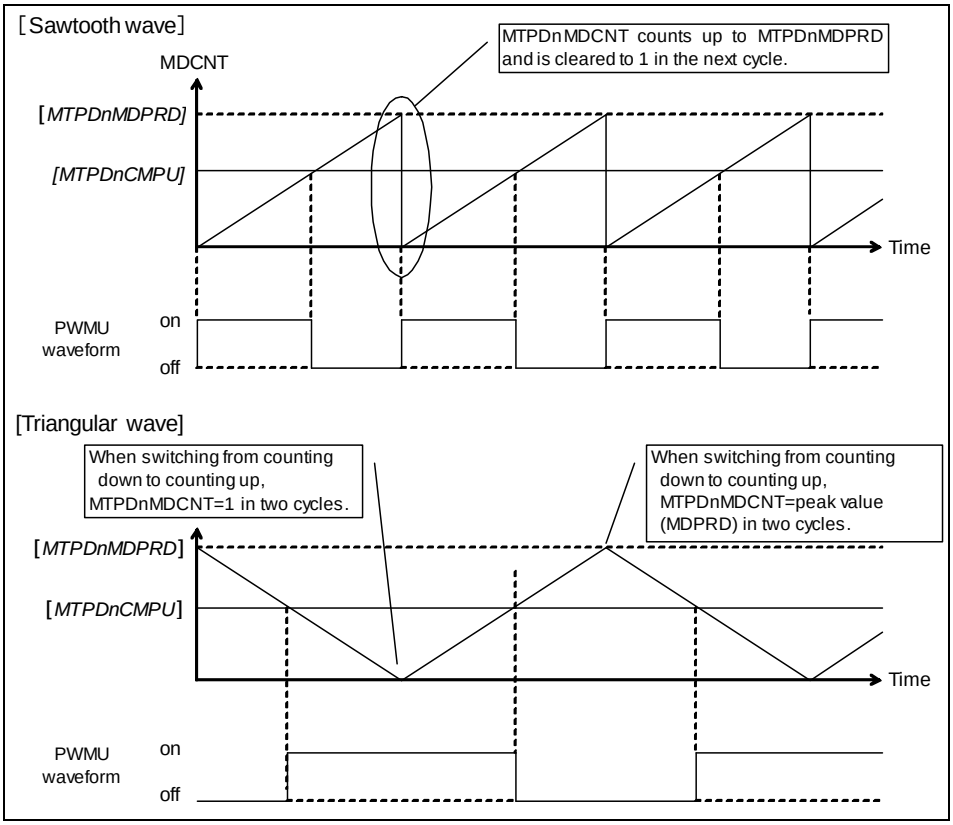


Fig 10-4 PWM Waveforms

(3) Waveform mode

3-phase PWM waveforms can be generated in the following two modes:

i) 3-phase independent mode:
Each of the PWM compare registers for the three phases is set independently to generate independent PWM waveforms for each phase. This mode is used to generate drive waveforms such as sinusoidal waves.

ii) 3-phase common mode:
Only the U-phase PWM compare register is set to generate identical PWM waveforms for all the three phases. This mode is used for rectangular wave drive of brushless DC motors.

(4) Interrupt processing

The pulse width modulation circuit generates PWM interrupt requests in synchronization with PWM waveforms. The PWM interrupt period can be set to half a PWM period, one PWM period, two PWM periods or four PWM periods.

- PMD Control Register (MTPDnMDCR)

0x4005 0408, 0x4005 0488

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
Read/Write	R→0	R/W						
After reset	0	0	0	0	0	0	0	0

<PWMMD>: PWM carrier waveform

0: PWM mode 0 (edge-aligned PWM, sawtooth wave)

1: PWM mode 1(center-aligned PWM, triangular wave)

This bit selects the PWM mode. PWM mode 0 is edge-aligned PWM and PWM mode 1 is center-aligned PWM.

<INTPRD>: PWM interrupt period

00: Interrupt request at every 0.5 PWM period (PWM mode 1 only)

01: Interrupt request at every PWM period

10: Interrupt request at every 2 PWM periods

11: Interrupt request at every 4 PWM periods

This field selects the PWM interrupt period from 0.5 PWM period, one PWM period, two PWM periods and four PWM periods.

When <INTPRD>=00, the contents of the compare registers (MTPDnCMPU, MTPDnCMPV, MTPDnvCMPW) and period register (MTPDnMDPRD) are updated into their respective buffers when the internal counter equals 1 or the MTPDnMDPRD value.

<PINT>: PWM interrupt timing

0: Interrupt request when PWM counter = 1

1: Interrupt request when PWM counter = MTPDnMDPRD

This bit selects whether to generate an interrupt request when the PWM counter equals its minimum or maximum value. When the edge-aligned PWM mode is selected, an interrupt request is generated when the PWM counter equals the MTPDnMDPRD value. When the PWM interrupt period is set to every 0.5 PWM period, an interrupt request is generated when the PWM counter equals 1 or MTPDnMDPRD.)

<DTYMD>: Duty mode

0: 3-phase common mode

1: 3-phase independent mode

This bit selects whether to make duty setting independently for each phase or to use the CMPU register for all three phases.

<SYNTMD>: Port output mode

This bit specifies the port output setting of the U, V and W phases. (See Table 10-8)

<PWMCK>: PWM period extension mode

0: Normal period

1: 4x periods

When <PWMCK>=0, the PWM counter operates with a resolution of 25 ns at fsys=40 MHz.

* Sawtooth wave: 25ns, triangular wave: 50 ns

When <PWMCK>=1, the PWM counter operates with a resolution of 100 ns at fsys=40 MHz.

* Sawtooth wave: 100 ns, triangular wave: 200 ns

- PWM Counter Status Register (MTPDnCNTSTA)

0x4005 040C, 0x4005 048C

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

<UPDWN>: PWM counter flag

0: Up-counting

1: Down-counting

This bit indicates whether the PWM counter is up-counting or down-counting.

When the edge-aligned PWM mode is selected, this bit is always read as 0.

- PWM Counter Register (MTPDnMDCNT)

0x4005 0410, 0x4005 0490

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

<MDCNT>: PWM counter

PMD counter value (resolution: 25 ns at fsys = 40 MHz)

* Sawtooth wave: 25 ns, triangular wave: 50 ns

* When MTPDnMDCR<PWMCK>=1, the counter resolution becomes 100 ns.

A16-bit counter for reading the PWM period count value. It is read-only.

- PWM Period Register (MTPDnMDPRD)

0x4005 0414, 0x4005 0494

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	MDPRD							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	MDPRD							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<MDPRD>: PWM period

MDPRD≧0x0010

A 16-bit register for specifying the PWM period. This register is double-buffered and can be changed even when the PWM counter is operating. The buffer is loaded at every PWM period. (That is, when the PWM counter matches the MDPRD value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or MDPRD. The least significant bit must be set as 0.)

If <MDPRD> is set to a value less than 0x0010, it is automatically assumed to be 0x0010. (The register retains the actual value that is written.)

* Do not write to this register in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

- PWM Compare Registers (MTPDnCMPU, MTPDnCMPV, MTPDnCMPW)

MPT01(0x4005 0418-041B),MPT12(0x4005 0498-049B)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	CMPU0,1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	CMPU0,1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

MPT0(0x4005 041C-041F),MPT1(0x4005 049C-049F)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	CMPV0,1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	CMPV0,1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

MPT0(0x4005 0420-0423),MPT1(0x4005 04A0-04A3)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	CMPW0,1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	CMPW0,1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<CMPU, CMPV, CMPW>: PWM pulse width

Compare registers (resolution: 25 ns at fsys =40 MHz)

* Sawtooth wave: 25 ns, triangular wave: 50 ns
* When MTPDnMDCR<PWMCK>=1, the counter resolution becomes 100 ns.

CMPU, CMPV and CMPW are compare registers for determining the output pulse width of the U, V and W phases. Theses registers are double-buffered. Pulse width is determined by comparing the buffer and the PWM counter to evaluate which is larger. (To be loaded when the PWM counter value matches the MDPRD value. When 0.5 PWM period is selected, loading is performed when the PWM counter matches 1 or MDPRD.) When this register is read, the value of the first buffer (data set via the bus) is returned.

* Do not write to these registers in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

10.7.3.2 Conduction Control Circuit

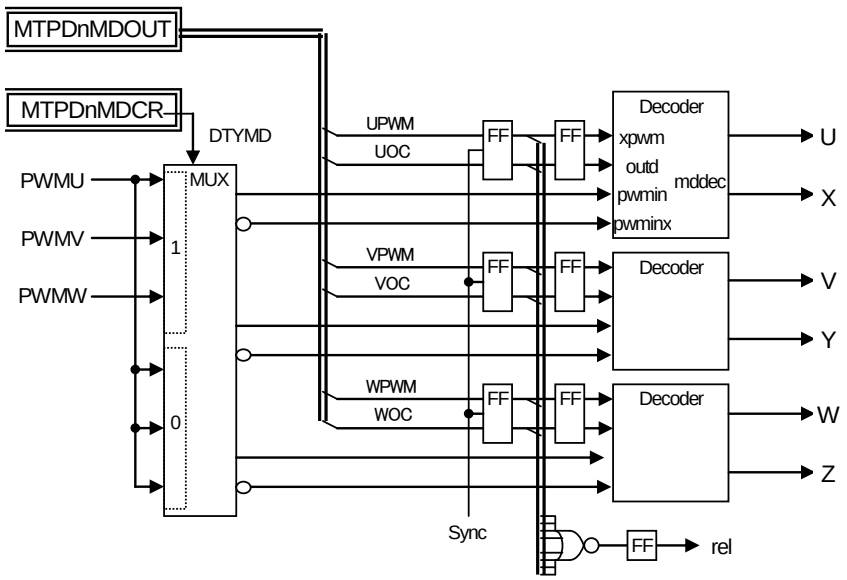


Fig 10-5 Conduction Control Circuit

The conduction control circuit performs output port control according to the settings made in the PMD output register (MTPDnMDOUT). The MDOUT register bits are divided into two parts: settings for the synchronizing signal for port output and settings for port output. The latter part is double-buffered and update timing can be set as synchronous or asynchronous to PWM.

The output settings for six port lines are made independently for each of the upper and lower phases through the bits 10 to 8 of the MDOUT register and bits 3 and 2 of the MTPDnMDPOT register. In addition, bits 10 to 8 of the MTPDnMDOUT register select PWM or H/L output for each of the U, V and W phases. When PWM output is selected, PWM waveforms are output. When H/L output is selected, output is fixed to either a high or low level. Table 10-8 shows a summary of port outputs according to port output settings in the MTPDnMDOUT register and polarity settings in the MTPDnMDCR register.

- PMD Output Setting Register (MTPDnMDPOT)

0x4005 042C, 0x4005 04AC

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	POLH	POLL	PSYNCS	
Read/Write	R→0	R→0	R→0	R→0	R/W	R/W	R/W	
After reset	0	0	0	0	0	0	0	0

<PSYNCS>: MDOUT transfer timing

00: Async to PWM
01: Load when PWM counter = 1
10: Load when PWM counter= MDPRD
11: Load when PWM counter = 1 or MDPRD

PSYNCS selects the timing when the U-, V- and W-phase output settings are reflected in port outputs
(sync or async to the PWM counter peak, bottom or peak/bottom).
* This field must be set while MTPDnMDEN<PWMEN>=0.

<POLL>: Lower phase port polarity

0: Active low
1: Active high

POLL selects the output port polarity of the lower phases.
* This bit must be set while MTPDnMDEN<PWMEN>=0.

<POLH>: Upper phase port polarity

0: Active low
1: Active high

POLH selects the output port polarity of the upper phases.
* This bit must be set while MTPDnMDEN<PWMEN>=0.

- PMD Output Control Register (MTPDnMDOUT)

0x4005 0428, 0x4005 04A8

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	WPWM	VPWM	UPWM
Read/Write	R→0	R→0	R→0	R→0	R→0	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	WOC		VOC		UOC	
Read/Write	R→0	R→0	R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

<UOC, VOC, WOC>,<UPWM, VPWM,WPWM>: U-, V-, and W-phase output control

The MTPDnMDOUT register controls the port outputs of the U, V and W phases (see Table 10-8 below.)

* To load the second buffer of MTPDnMDOUT with a value updated via the bus, select the bus mode (default) by setting MTPDnMODESEL<MDSEL> to 0.

* Do not write to this register in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.

Table 10-8 Port Outputs according to the UOC, VOC, WOC, UPWM, VPWM and WPWM Settings

○ MTPDnMDCR<SYNTMD>=0

Polarity: Active high (MDPOT bits 3, 2=1)

MDOUT Output Control		MDOUT bits 10, 9, 8 H/L or PWM Output Select			
Bits 5, 3, 1 Upper phase	Bits 4, 2, 0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	/PWM	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	/PWM

Polarity: Active low (MDPOT bits 3, 2=0)

MDOUT Output Control		MDOUT bits 10, 9, 8 H/L or PWM Output Select			
Bits 5,3,1 Upper phase	Bits 4,2,0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	/PWM
0	1	H	L	H	/PWM
1	0	L	H	/PWM	H
1	1	L	L	/PWM	PWM

○ MTPDnMDCR<SYNTMD>=1

Polarity: Active high (MDPOT bits 3, 2=1)

MDOUT Output Control		MDOUT bits 10, 9, 8 H/L or PWM Output Select			
Bits 5, 3, 1 Upper phase	Bits 4, 2, 0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	L	L	/PWM	PWM
0	1	L	H	L	/PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	/PWM

Polarity: Active low (MDPOT bits 3, 2=0)

MDOUT Output Control		MDOUT bits 10, 9, 8 H/L or PWM Output Select			
Bits 5,3,1 Upper phase	Bits 4,2,0 Lower phase	0: H/L output		1: PWM output	
		Upper phase output	Lower phase output	Upper phase output	Lower phase output
0	0	H	H	PWM	/PWM
0	1	H	L	H	PWM
1	0	L	H	/PWM	H
1	1	L	L	/PWM	PWM

Output Settings for Center-off PWM

Center-off PWM can be supported by the following settings.

Table 10-9 Register Settings for Center-off PWM

	Normal PWM center on	U-Phase PWM center off	V-Phase PWM center off	W-Phase PWM center off
CMPU	duty_U	MDPRD-duty_U	duty_U	duty_U
CMPV	duty_V	duty_V	MDPRD-duty_V	duty_V
CMPW	Duty_W	duty_W	duty_W	MDPRD-duty_W
UOC	11	00	11	11
VOC	11	11	00	11
WOC	11	11	11	00

10.7.3.3 Protection Control Circuit

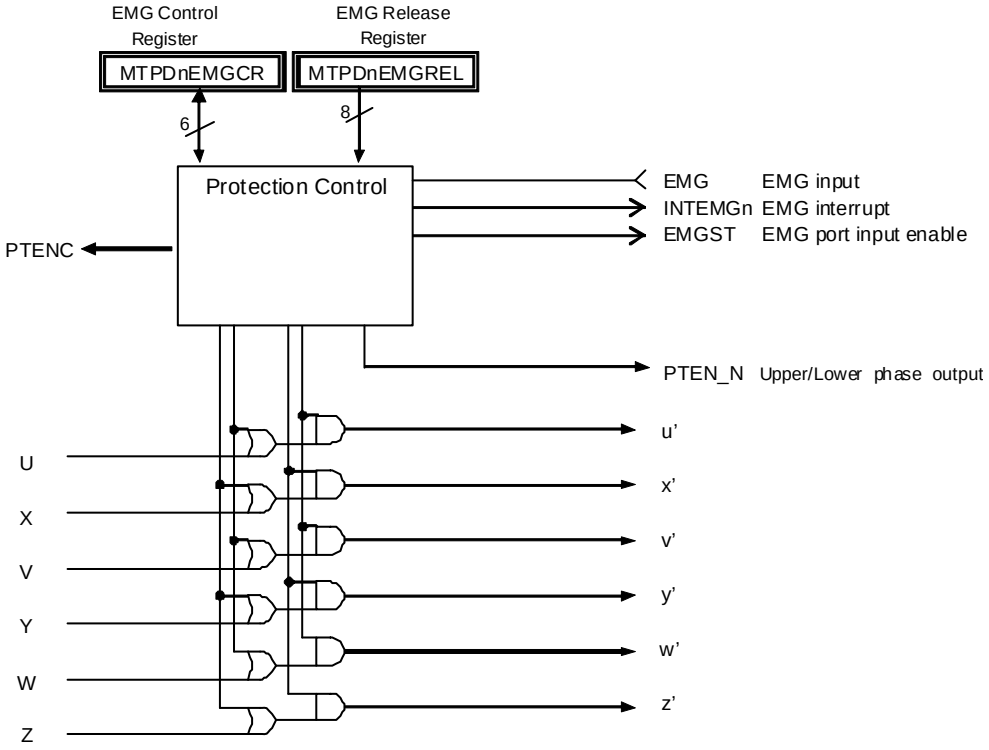


Fig 10-6 Protection Control Circuit

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

(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 MTPDnEMGCR<EMGMD> setting) and an EMG interrupt (INTEMG) is generated. MTPDnEMGCR<EMGMD> can be set to output a control signal that sets external output ports to high impedance in case of an emergency.

A tool break also disables all six PWM output lines depending on the EMGCR<EMGMD> setting. When a tool break occurs, external output ports can be set to high impedance through the setting of the PORTMD register.

EMG protection is set through the EMG Control Register (MTPDnEMGCR). A read value of 1 in MTPDnEMGSTA<EMGST> indicates that the EMG protection circuit is active. In this state, EMG protection can be released by setting all the port output lines inactive (MDOUT[10:8][5:0]) and then setting MTPDnEMGCR<EMGRS> to 1.

To disable the EMG protection function, write 0x5A and 0xA5 in this order to the MTPDnEMGREL register and then clear MTPDnEMGCR<EMGEN> to 0. (These three instructions must be executed consecutively.) While the EMG protection input is low, any attempt to release the EMG protection state is ignored. Before setting MTPDnEMGCR<EMGRS> to 1 to release EMG protection, make sure that MTPDnEMGST<EMGI> is high.

The EMG protection circuit can be disabled only after the specified key codes (0x5A, 0xA5) are written in the MTPDnEMGREL register to prevent it from being inadvertently disabled.

- EMG Release Register (MTPDnEMGREL)

0x4005 0430, 0x4005 04B0

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	EMGREL							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

<EMGREL>: EMG disable code

The EMG protection functions can be disabled by setting 5A and A5 in this order to bits 7 to 0 of the MTPDnEMGREL register.

When disabling these functions, MTPDnEMGCR<EMGEN> must be cleared to 0.

* This register is used for both the EMG function.

- EMG Control Register (MTPDnEMGCR)

0x4005 0434, 0x4005 04B4

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	EMGCNT			
Read/Write	R→0	R→0	R→0	R→0	R/W			
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	INHEN	EMGMD		-	EMGRS	EMGEN
Read/Write	R→0	R→0	R/W	R/W		R/W	W	R/W
After reset	0	0	1	1	1	0	0	1

<EMGEN>: EMG protection circuit enable/disable

0: Disable

1: Enable

The EMG protection circuit is enabled by setting this bit to 1. In the initial state, the EMG protection circuit is enabled.

To disable this circuit, write 5A and A5 in this order to the MTPDnEMGREL register and then clear the EMGEN bit to 0.

<EMGRS>: EMG protection release

0: -

1: Release protection

EMG protection can be released by setting the MTPDnMDOUT register to 0 and then setting the EMGRS bit

to 1.

This bit is always read as 0.

* Be sure to write 0 to both the upper bits [10:8] and lower bits [5:0].
* Before releasing EMG protection, make sure that the EMG input has returned to high.

Bit2: write “0”

<EMGMD>: EMG protection mode select

00: PWM output control disabled / Port output = All phases High-Z

01: All upper phases ON, all lower phases OFF / Port output = Lower phases High-Z

10: All upper phases OFF, all lower phases ON / Port output = Upper phases High-Z

11: All phases OFF / Port output = All phases High-Z

* ON = PWM output (no output control), OFF = Low [when <POLL>,<POLH>=1 (active high)]

This field controls PWM output and port output of the upper and lower phases in case of an emergency.

<INHEN>: Tool break enable/disable

0: Disable

1: Enable

This bit selects whether or not to stop the PMD when the PMD stop signal is input from the tool. In the initial state, tool breaks are enabled.

<EMGCNT>: EMG input detection time

0 to 15 (When <EMGCNT=0, the noise filter is bypassed.)

EMGCNT×16/fsys (resolution: 400[ns] at 40 MHz)

- EMG Status Register (MTPDnEMGSTA)

0x4005 0438, 0x4005 04B8

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	EMGI	EMGST
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R	R
After reset	0	0	0	0	0	0	-	0

<EMGST>: EMG protection state

0: Normal operation

1: Protected

The EMG protection state can be known by reading this bit.

<EMGI>: EMG input

EMG protection state

The EMG input state can be known by reading this bit.

10.7.3.4 Dead Time Circuit

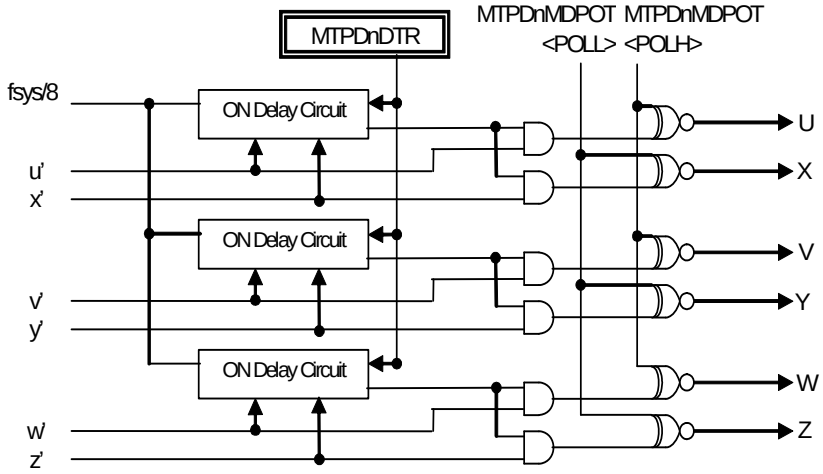


Fig 10-7 Dead Time Circuit

The dead time circuit consists of a dead time unit and an output polarity switching unit.

For each of the U, V and W phases, the ON delay circuit introduces a delay (dead time) when the upper and lower phases are switched to prevent a short circuit. The dead time is set to the Dead Time Register (DTR) as an 8-bit value with a resolution of 200 ns at 40 MHz.

The output polarity switching circuit allows the polarity (active high or active low) of the upper and lower phases to be independently set through MDPOT<POLH> and <POLL>.

- Dead Time Register (MTPDnDTR)

0x4005 0444, 0x4005 04C4

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	DTR							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<DTR>: Dead time

200 nsec × 8 bits (up to 51 μs at fsys = 40 MHz)

* Do not change this register while MTPDnMDEN<PWMEN>=1.

10.7.3.5 Sync Trigger Generation Circuit

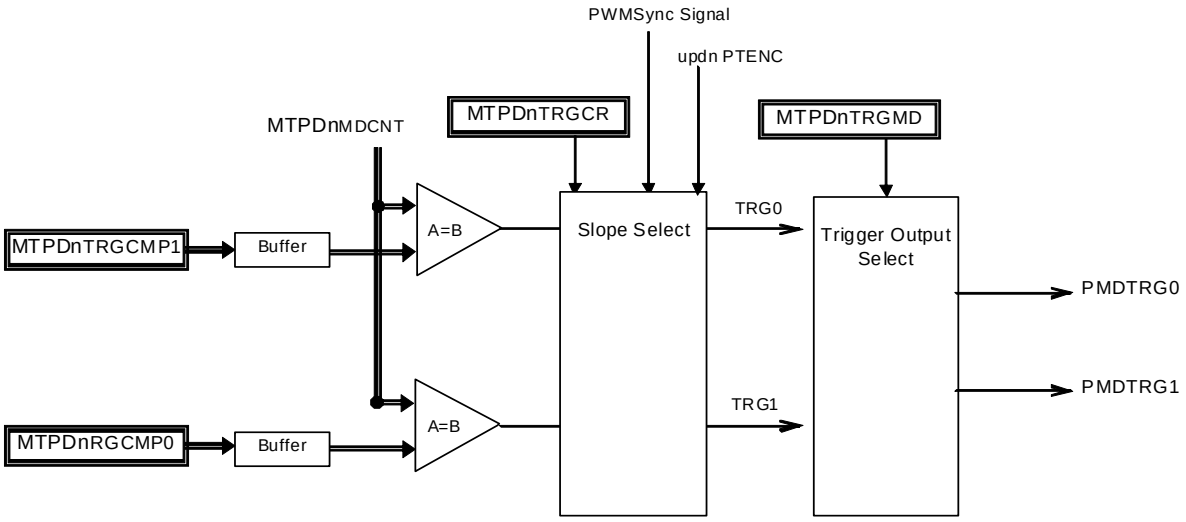


Fig 10-8 Sync Trigger Generation Circuit

The sync trigger generation circuit generates trigger signals for starting ADC sampling in synchronization with PWM. The ADC trigger signal (PMDTRG) is generated by a match between MTPDnMDCNT and MTPDnTRGCMP. The signal generation timing can be selected from up-count match, down-count match and up-/down-count match. When the edge-aligned PWM mode is selected, the ADC trigger signal is generated on an up-count match. When PWM output is disabled (MTPDnMDEN<PWMEN>=0), trigger output is also disabled.

- Trigger Compare Registers (MTPDnTRGCMP0, MTPDnTRGCMP1)

MPT0 (0x4005 0448-044B), MPT 1 (0x4005 04C8-04CB)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRGCMP0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRGCMP0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

MPT 0 (0x4005 044C-044F), MPT 1 (0x4005 04CC-04CF)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	TRGCMP1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRGCMP1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

<TRGCMP0-1>: Trigger output compare registers

When the PMD counter value (MDCNT) matches the value set in MTPDnTRGCMPx, PMDTRG is output. When MTPDnTRGCMPx is read, the value in the first buffer of the double buffers (data set via the bus) is returned.

MTPDnTRGCMPx should be set in a range of 1 to [MDPRD set value – 1].

* It is prohibited to set MTPDnTRGCMPx to 0 or the MTPDnMDPRD value.
* To load the data in MTPDnTRGCMP0 and MTPDnTRGCMP1 to the second buffers, select the bus mode (default) by setting MTPDnMODESEL<MDSEL> to 0.
* Do not write to these registers in byte units. If the upper 8 bits [15:8] and the lower 8 bits [7:0] are written separately, operation cannot be guaranteed.
* When MTPDnTRGCMPx is set to 0x0001, no trigger output is made only in the first cycle after PWM start (MTPDnMDEN<PWMEN>1).

Update Timing of the Trigger Compare Register (MTPDnTRGCMPx)

The Trigger Compare Register (MTPDnTRGCMPx) is double-buffered. The timing at which the data written to MTPDnTRGCMPx is loaded to the second buffer depends on the setting of MTPDnTRGCR<TRGxMD>. When MTPDnTRGCR<TRGxBE> is set to 1, data written to MTPDnTRGCMPx is immediately loaded to the second buffer.

Table 10-10 TRGCMPx Buffer Update Timing according to Trigger Output Mode Setting

TRGxMD	MTUFx Update Timing
000:Trigger output disabled	Always updated
001:Trigger output on down-count match	Updated when PWM counter equals MDPRD (PWM carrier peak)
010:Trigger output on up-count match	Updated when PWM counter equals 1 (PWM carrier bottom)
011:Trigger output on up-/down-count match	Updated when PWM counter equals 1 or MDPRD (PWM carrier peak/bottom)
100:Trigger output at PWM carrier peak	Always updated
101:Trigger output at PWM carrier bottom	
110:Trigger output at PWM carrier peak/bottom	
111:Trigger output disabled	

- Trigger Control Register (MTPDnTRGCR)

0x4005 0458, 0x4005 04D8

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-			-	-		
Read/Write	R/W	R/W			R/W	R/W		
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
Read/Write	R/W	R/W			R/W	R/W		
After reset	0	0	0	0	0	0	0	0

<TRG0MD, TRG1MD >: PMDTRG0 to PMDTRG1 mode setting

000: Trigger output disabled
001: Trigger output at down-count match
010: Trigger output at up-count match
011: Trigger output at up-/down-count match
100: Trigger output at PWM carrier peak
101: Trigger output at PWM carrier bottom
110: Trigger output at PWM carrier peak/bottom
111: Trigger output disabled

This register selects trigger output timing.
When the PMD is set to the edge-aligned mode, trigger outputs are made on up-count match or at PWM carrier peak even if down-count match or PWM carrier bottom is selected.

* When <TRGxMD>=011, TRGCMPx=0x0001 and MDCR<PWMMD>=1 (triangular wave), one trigger output is made per period.

< TRG0BE、TRG1BE、TRG2BE、TRG3BE >: PMDTRG0 to PMDTRG1 buffer update timing

0: Sync
1: Async (The value written to PMDTRGx is immediately reflected.)

This bit enables asynchronous updating of the PMDTRG0 to PMDTRG1 buffers.

Bit[15:7] : write “00000000”

- Trigger Output Mode Setting Register (MTPDnTRGMD)

0x4005 045C, 0x4005 04DC

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R→0	R→0
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	EMGTGE
Read/Write	R→0	R→0	R→0	R→0	R→0	R→0	R/W	R/W
After reset	0	0	0	0	0	0	0	0

<EMGTGE>: Output enable in EMG protection state

0: Disable trigger output in the protection state

1: Enable trigger output in the protection state

This bit enables or disables trigger output in the EMG protection state.

Bit1: write “0”

11 12/10-Bit Analog-to-Digital Converters

Important
TMPM382 (64-pin version) have 10 analog inputs. The AIN0 to AIN9 are available.

The TMPM380 contains a 12/10(selectable)-bit successive-approximation analog-to-digital converter (ADC).
The ADC has 18 analog inputs.

Functions and features

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

11.1 Block Diagram

The following shows a block diagram of the ADC.

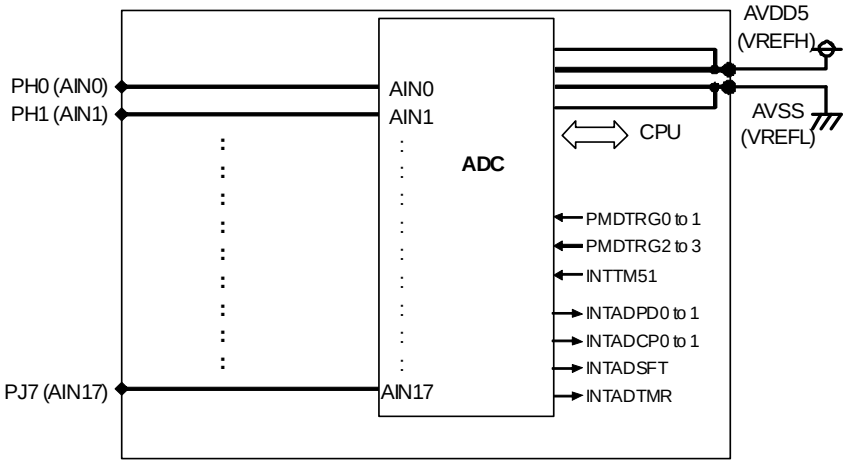


Fig 11.1 AD converters Block Diagram

Note : TMPM382 has 10 analog inputs. The AIN0 to AIN9 are available.
TMPM382 doesn't have two PMD trigger signal (PMDTRG2,PMDTRG3).

11.2 List of Registers

The ADCs have the following registers.

Address	Register Name	Description
4003_00_00	ADCLK	ADC Clock Setting Register
4003_00_04	ADMOD0	ADC Mode Setting Register 0
4003_00_08	ADMOD1	ADC Mode Setting Register 1
4003_00_0C	ADMOD2	ADC Mode Setting Register 2
4003_00_10	ADCMPCR0	Monitoring Setting Register 0
4003_00_14	ADCMPCR1	Monitoring Setting Register 1
4003_00_18	ADCMP0	AD Conversion Result Compare Register 0
4003_00_1C	ADCMP1	AD Conversion Result Compare Register 1
4003_00_20	ADREG0	AD Conversion Result Register 0
4003_00_24	ADREG1	AD Conversion Result Register 1
4003_00_28	ADREG2	AD Conversion Result Register 2
4003_00_2C	ADREG3	AD Conversion Result Register 3
4003_00_30	ADREG4	AD Conversion Result Register 4
4003_00_34	ADREG5	AD Conversion Result Register 5
4003_00_38	ADREG6	AD Conversion Result Register 6
4003_00_3C	ADREG7	AD Conversion Result Register 7
4003_00_40	ADREG8	AD Conversion Result Register 8
4003_00_44	ADREG9	AD Conversion Result Register 9
4003_00_48	ADREG10	AD Conversion Result Register 10
4003_00_4C	ADREG11	AD Conversion Result Register 11
4003_00_50	ADPSEL0	PMD Trigger Program Number Select Register 0
4003_00_54	ADPSEL1	PMD Trigger Program Number Select Register 1
4003_00_58	ADPSEL2	PMD Trigger Program Number Select Register 2
4003_00_5C	ADPSEL3	PMD Trigger Program Number Select Register 3
4003_00_80	ADPINTS0	PMD Trigger Interrupt Select Register 0
4003_00_84	ADPINTS1	PMD Trigger Interrupt Select Register 1
4003_00_88	ADPINTS2	PMD Trigger Interrupt Select Register 2
4003_00_8C	ADPINTS3	PMD Trigger Interrupt Select Register 3
4003_00_90	ADPINTS4	PMD Trigger Interrupt Select Register 4
4003_00_94	ADPINTS5	PMD Trigger Interrupt Select Register 5
4003_00_98	ADPSET0	PMD Trigger Program Register 0
4003_00_9C	ADPSET1	PMD Trigger Program Register 1
4003_00_A0	ADPSET2	PMD Trigger Program Register 2
4003_00_A4	ADPSET3	PMD Trigger Program Register 3
4003_00_A8	ADPSET4	PMD Trigger Program Register 4
4003_00_AC	ADPSET5	PMD Trigger Program Register 5
4003_00_B0	ADTSET03	Timer Trigger Program Registers 0 to 3
4003_00_B4	ADTSET47	Timer Trigger Program Registers 4 to 7
4003_00_B8	ADTSET811	Timer Trigger Program Registers 8 to 11
4003_00_BC	ADSSET03	Software Program Registers 0 to 3
4003_00_C0	ADSSET47	Software Program Registers 4 to 7
4003_00_C4	ADSSET811	Software Program Registers 8 to 11
4003_00_C8	ADASET03	Constant Conversion Program Registers 0 to 3
4003_00_CC	ADASET47	Constant Conversion Program Registers 4 to 7
4003_00_D0	ADASET811	Constant Conversion Program Registers 8 to 11
4003_00_D4	ADMOD3	ADC Mode Setting Register 3

Table 11.1 AD conversion registers

11.3 Register Descriptions

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

11.3.1 ADC Clock Setting Register (ADCLK)

ADCLK
0x4003_0000

	7	6	5	4	3	2	1	0
Bit symbol	-	TSH3	TSH2	TSH1	TSH0	ADCLK2	ADCLK1	ADCLK0
Read/Write	R	R/W				R/W		
After reset	0	1011				000		
Function	Always read as 0.	Write ”1001”				AD prescaler output (SCLK) select 000: fc 001: Reserved 010: Reserved 011: Reserved 1XX: Reserved		

Note1: AD conversion time(T) is as follow;
12-bit mode : T=74*(1/SCLK), 10-bit mode : T=68*(1/SCLK)

11.3.2 Mode Setting Registers

The ADC Mode Setting Registers (ADMOD0, ADMOD1, ADMOD2 and ADMOD3) are used to select how AD conversion is started.

11.3.2.1 ADMOD0

ADMOD0
0x4003_0004

	7	6	5	4	3	2	1	0
Bit symbol	-						DACON	ADSS
Read/Write	R						R/W	W
After reset	0						0	0
Function	Always read as 0.						DAC control 0: off 1: On	Software triggered conversion 0 : Don't care 1: Start

Setting <DACON> to “1”, when using the ADC.
Setting <ADSS> to “1” starts AD conversion (software triggered conversion). Receiving trigger signal from PMD(MPT) or TMRB(interrupt) starts AD conversion also.
For detail setting, please read the chapter about PMD(MPT) and TMRB.

11.3.2.2 ADMOD1

ADMOD1
0x4003_0008

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

Setting <ADEN> to “1”, when using the ADC. After Setting <ADEN> to “1”, setting <ADAS> to “1” starts AD conversion and repeat conversion.

11.3.2.3 ADMOD2

ADMOD2
0x4003_000C

	7	6	5	4	3	2	1	0
Bit symbol	-						ADSFN	ADBFN
Read/Write	R						R	R
After reset	0						0	0
Function	Always read as 0.						Software conversion busy flag 0: Conversion completed 1: Conversion in progress	AD conversion busy flag 0: Conversion not in progress 1: Conversion in progress

The <ADBFN> is an AD conversion busy flag. When AD conversion is started regardless of conversion factor (PMD, Timer, Software, Constant), <ADBFN> is set to “1”. When finished AD conversion, <ADBFN> is cleared to “0”.
The <ADSFN> is a software AD conversion busy flag. After <ADSS> was set to “1”, when AD conversion is actually started, <ADSFN> is set to “1”. When finished AD conversion, <ADSFN> is cleared to “0”.

11.3.2.4 ADMOD3

ADMOD3

ADMOD3
0x4003_00D4

	7	6	5	4	3	2	1	0
Bitsymbol	-	-	PMODE2	PMODE1	PMODE0	-	-	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	1	0	1	1	0	0	0
Function	Write same value as initial value.		Write “100”			Write same value as initial value.		

	15	14	13	12	11	10	9	8
Bitsymbol	-	-	-	-	BITS1	BITS0		RCUT
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	1	0	1
Function	Write same value as initial value.				bit resolution selector 00:10bit 01:12bit 10:Reserved 11:Reserved		Write “0”	Low power mode select 0:Normal 1:Low power

Note : <PMODE[2:0]> must be set to “100”. And do not change other bits<bit15 to12,9,7,6,2 to0> in ADMOD3 register.

The ADC can select the resolution of 10bits or 12bits. Setting the <BITS1>/<BITS0> on ADMOD3 register.

The ADC can decrease the current consumption when stopping. Setting the <RCUT> to “1” enables the ADMOD3 register.

11.3.3 Monitoring Setting Registers

The ADCs have the AD conversion result monitoring function.

11.3.3.1 ADCMPCR0,ADCMPCR1

The ADCMPCR0 and ADCMPCR1 registers are used to enable or disable comparison between an AD conversion result and the specified comparison value, to select the register to be compared with an AD conversion result and to set how many times comparison should be performed to determine the result.

After fixing the conversion result, The interrupt signal(INTADCP0,INTADCP1) is generated.

ADCMPCR0

ADCMPCR0
0x0043_0010

	7	6	5	4	3	2	1	0
Bit symbol	CMP0EN	-	-	ADBIG0	REGS03	REGS02	REGS01	REGS00
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Monitoring function 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Comparison condition 0: Larger than or equal to compare register 1: Smaller than or equal to compare register	AD conversion result register to be compared 0000: ADREG0 0100: ADREG4 1000: ADREG8 0001: ADREG1 0101: ADREG5 1001: ADREG9 0010: ADREG2 0110: ADREG6 1010: ADREG10 0011: ADREG3 0111: ADREG7 1011: ADREG11			
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	CMPCNT03	CMPCNT02	CMPCNT01	CMPCNT00
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Comparison count for determining the result 0000: After every comparison 0001: After two comparisons . . 1111: After 16 comparisons			

ADCMPCR1

ADCMPCR1
0x4003_0014

	7	6	5	4	3	2	1	0
Bit symbol	CMP1EN	-	-	ADBIG1	REGS13	REGS12	REGS11	REGS10
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Monitoring function 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Comparison condition 0: Larger than or equal to compare register 1: Smaller than or equal to compare register	AD conversion result register to be compared 0000: ADREG0 0100: ADREG4 1000: ADREG8 0001: ADREG1 0101: ADREG5 1001: ADREG9 0010: ADREG2 0110: ADREG6 1010: ADREG10 0011: ADREG3 0111: ADREG7 1011: ADREG11			
	15	14	13	12	11	10	9	8
Bit symbol	-	-	-	-	CMPCNT13	CMPCNT12	CMPCNT11	CMPCNT10
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Comparison count for determining the result 0000: After every comparison 0001: After two comparisons . . 1111: After 16 comparisons			

11.3.3.2 Conversion Result Compare Register

The ADCMP0 and ADCMP1 registers specify the value to be compared with an AD conversion result. The upper 12 bits (bits 4 to 15) are used.

ADCMP0

ADCMP0
0x4003_0018

	7	6	5	4	3	2	1	0
Bit symbol	AD0CMP03	AD0CMP02	AD0CMP01	AD0CMP00	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of the value to be compared with an AD conversion result				Always read as 0.			
	15	14	13	12	11	10	9	8
Bit symbol	AD0CMP11	AD0CMP10	AD0CMP09	AD0CMP08	AD0CMP07	AD0CMP06	AD0CMP05	AD0CMP04
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of the value to be compared with an AD conversion result							

ADCMP1

ADCMP1
0x4003_001C

	7	6	5	4	3	2	1	0
Bit symbol	AD1CMP03	AD1CMP02	AD1CMP01	AD1CMP00	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of the value to be compared with an AD conversion result				Always read as 0.			
	15	14	13	12	11	10	9	8
Bit symbol	AD1CMP11	AD1CMP10	AD1CMP09	AD1CMP08	AD1CMP07	AD1CMP06	AD1CMP05	AD1CMP04
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of the value to be compared with an AD conversion result							

11.3.4 AD Conversion Result Registers

11.3.4.1 ADREG0 to ADREG11

The ADREGn (n = 0 to 11) register is used to store the result of an AD conversion. Bit 0 (ADRnRF) is a flag that is set when an AD conversion result is stored in the ADREGn register and is cleared when the low-order byte of ADREGn is read. Bit 1 (OVRn) is an overrun flag. This flag is set when a new AD conversion result is stored before the low-order byte of ADREGn is read and is cleared when the low-order byte of ADREGn is read.

There are twelve ADREGn registers, which are all functionally equivalent.

ADREG0

ADREG0
0x4003_0020

	7	6	5	4	3	2	1	0
Bit symbol	ADR003	ADR002	ADR001	ADR000	-	-	OVR0	ADR0RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR011	ADR010	ADR009	ADR008	ADR007	ADR006	ADR005	ADR004
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG1

ADREG1
0x4003_0024

	7	6	5	4	3	2	1	0
Bit symbol	ADR103	ADR102	ADR101	ADR100	-	-	OVR0	ADR1RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR111	ADR110	ADR109	ADR108	ADR107	ADR106	ADR105	ADR104
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG2

ADREG2
0x4003_0028

	7	6	5	4	3	2	1	0
Bit symbol	ADR203	ADR202	ADR201	ADR200	-	-	OVR0	ADR2RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR211	ADR210	ADR209	ADR208	ADR207	ADR206	ADR205	ADR204
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG3

ADREG3
0x4003_002C

	7	6	5	4	3	2	1	0
Bit symbol	ADR303	ADR302	ADR301	ADR300	-	-	OVR0	ADR3RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR311	ADR310	ADR309	ADR308	ADR307	ADR306	ADR305	ADR304
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG4

ADREG4
0x4003_0030

	7	6	5	4	3	2	1	0
Bit symbol	ADR403	ADR402	ADR401	ADR400	-	-	OVR0	ADR4RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR411	ADR410	ADR409	ADR408	ADR407	ADR406	ADR405	ADR404
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG5

ADREG5
0x4003_0034

	7	6	5	4	3	2	1	0
Bit symbol	ADR503	ADR502	ADR501	ADR500	-	-	OVR0	ADR5RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR511	ADR510	ADR509	ADR508	ADR507	ADR506	ADR505	ADR504
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG6

ADREG6
0x4003_0038

	7	6	5	4	3	2	1	0
Bit symbol	ADR603	ADR602	ADR601	ADR600	-	-	OVR0	ADR6RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR611	ADR610	ADR609	ADR608	ADR607	ADR606	ADR605	ADR604
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG7

ADREG7
0x4003_003C

	7	6	5	4	3	2	1	0
Bit symbol	ADR703	ADR702	ADR701	ADR700	-	-	OVR0	ADR7RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR711	ADR710	ADR709	ADR708	ADR707	ADR706	ADR705	ADR704
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG8

ADREG6
0x4003_0040

	7	6	5	4	3	2	1	0
Bit symbol	ADR803	ADR802	ADR801	ADR800	-	-	OVR0	ADR8RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR811	ADR810	ADR809	ADR808	ADR807	ADR806	ADR805	ADR804
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG9

ADREG9
0x4003_0044

	7	6	5	4	3	2	1	0
Bit symbol	ADR903	ADR902	ADR901	ADR900	-	-	OVR0	ADR9RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR911	ADR910	ADR909	ADR908	ADR907	ADR906	ADR905	ADR904
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG10

ADREG10
0x4003_0048

	7	6	5	4	3	2	1	0
Bit symbol	ADR1003	ADR1002	ADR1001	ADR1000	-	-	OVR0	ADR10RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR1011	ADR1010	ADR1009	ADR1008	ADR1007	ADR1006	ADR1005	ADR1004
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

ADREG1 1

ADREG11
0x4003_004C

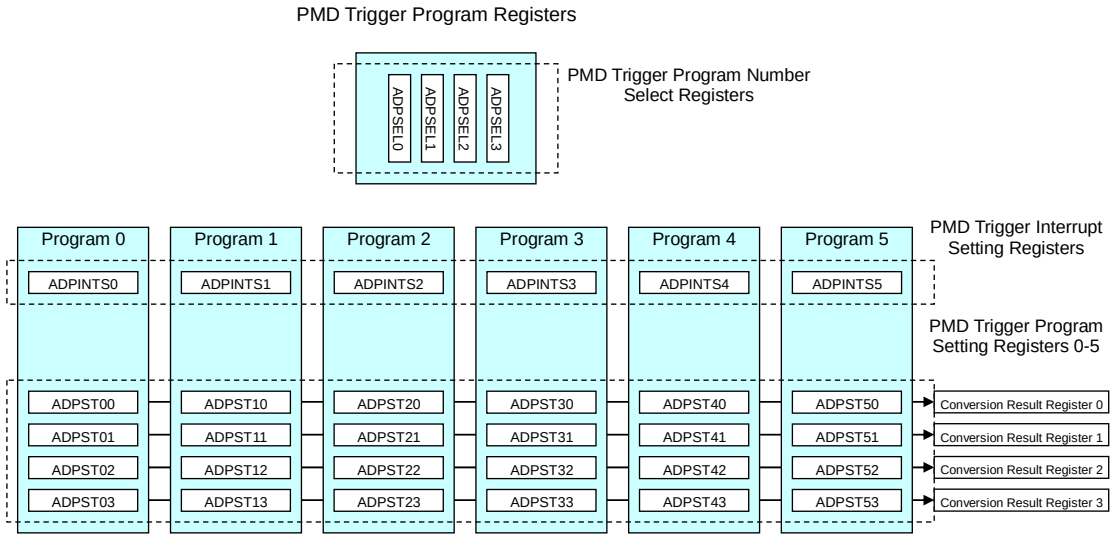
	7	6	5	4	3	2	1	0
Bit symbol	ADR1103	ADR1102	ADR1101	ADR1100	-	-	OVR0	ADR11RF
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 0 to 3 of an AD conversion result				Always read as 0.	Always read as 0.	Overrun flag 0: No overrun occurred 1: Overrun occurred	AD conversion result store flag 0: No result stored 1: Result stored
	15	14	13	12	11	10	9	8
Bit symbol	ADR1111	ADR1110	ADR1109	ADR1108	ADR1107	ADR1106	ADR1105	ADR1104
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Bits 4 to 11 of an AD conversion result							

11.3.5 PMD Trigger Program Registers

AD conversion can be started by a trigger from the PMD (MPT: PMD mode).

The PMD trigger program registers are used to specify the program to be started by each of four 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 (ADPSELp), PMD Trigger Interrupt Select Register (ADPINTSn) and PMD Trigger Program Setting Register (ADPSETnm). (p=0 to 3, n=0 to 5, m=0 to 3)



The PMD Trigger Program Number Select Register (ADPSELn) specifies the program to be started by each of four AD conversion start signals corresponding to four triggers generated by the PMD. Programs 0 to 5 are available.

The PMD Trigger Interrupt Select Register (ADPINTn) selects the interrupt to be generated upon completion of each program, and enables or disables the interrupt.

The PMD Trigger Program Setting Register (ADPSETnm) specifies the settings for each of programs 0 to 5. Each PMD Trigger Program Register is comprised of four registers for specifying the AIN input to be converted. The conversion results corresponding to the ADPSETn0 to ADPSETn3 registers are stored in the Conversion Result Registers 0 to 3 (ADREG0 to ADREG3).

11.3.5.1 PMD Trigger Program Number Select Registers

The PMD Trigger Program Number Select Registers (ADPSEL0 to ADPSEL03) select the program to be started (from among programs 0 to 5) by each of trigger inputs PMD0 to PMD3.

PMD Trigger Program Number Select Register 0 (ADPSEL0)

ADPSEL0
0x4003_0050

	7	6	5	4	3	2	1	0
Bit symbol	PENS0	-	-	-	-	PMDS02	PMDS01	PMDS00
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 1 (ADPSEL1)

ADPSEL1
0x4003_0054

	7	6	5	4	3	2	1	0
Bit symbol	PENS1	-	-	-	-	PMDS12	PMDS11	PMDS10
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 2 (ADPSEL2)

ADPSEL2
0x4003_0058

	7	6	5	4	3	2	1	0
Bit symbol	PENS2	-	-	-	-	PMDS22	PMDS21	PMDS20
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010 Program 2 110,111: Reserved 011: Program 3		

PMD Trigger Program Number Select Register 3 (ADPSEL3)

ADPSEL3
0x4003_005C

	7	6	5	4	3	2	1	0
Bit symbol	PENS3	-	-	-	-	PMDS32	PMDS31	PMDS30
Read/Write	R/W	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	PMD Trigger Program Select Register 3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	Always read as 0.	Always read as 0.	Program number select 000: Program 0 100: Program 4 001: Program 1 101: Program 5 010: Program 2 110, 111: Reserved 011: Program 3		

11.3.5.2 PMD Trigger Interrupt Select Registers

The PMD Trigger Interrupt Select Registers (ADPINTS0 to ADPINTS5) select the interrupt to be generated for each of programs 0 to 5.

ADPINTS0 (for program 0)

ADPINTS0
0x4003_0080

	7	6	5	4	3	2	1	0
Bit symbol	-						INTSEL01	INTSEL00
Read/Write	R						R/W	
After reset	0						0	
Function	Always read as 0.						Interrupt select 00: No interrupt output 01: INTADPD0 10: INTADPD1 11: No interrupt output	

ADPINTS1 (for program 1)

ADPINTS1
0x4003_0084

	7	6	5	4	3	2	1	0
Bit symbol	-						INTSEL11	INTSEL10
Read/Write	R						R/W	
After reset	0						0	
Function	Always read as 0.						Interrupt select 00: No interrupt output 01: INTADPD0 10: INTADPD1 11: No interrupt output	

ADPINTS2 (for program 2)

ADPINTS2
0x4003_0088

	7	6	5	4	3	2	1	0
Bit symbol	-						INTSEL21	INTSEL20
Read/Write	R						R/W	
After reset	0						0	
Function	Always read as 0.						Interrupt select 00: No interrupt output 01: INTADPD0 10: INTADPD1 11: No interrupt output	

ADPINTS3 (for program 3)

ADPINTS3
0x4003_008C

	7	6	5	4	3	2	1	0
Bit symbol	-						INTSEL31	INTSEL30
Read/Write	R						R/W	
After reset	0						0	
Function	Always read as 0.						Interrupt select 00: No interrupt output 01: INTADPD0 10: INTADPD1 11: No interrupt output	

ADPINTS4 (for program 4)

ADPINTS4
0x4003_0090

	7	6	5	4	3	2	1	0
Bit symbol	-						INTSEL41	INTSEL40
Read/Write	R						R/W	
After reset	0						0	
Function	Always read as 0.						Interrupt select 00: No interrupt output 01: INTADPD0 10: INTADPD1 11: No interrupt output	

ADINTS5 (for program 5)

ADPINS5
0x4003_0094

	7	6	5	4	3	2	1	0
Bit symbol	-						INTSEL51	INTSEL50
Read/Write	R						R/W	
After reset	0						0	
Function	Always read as 0.						Interrupt select 00: No interrupt output 01: INTADPD0 10: INTADPD1 11: No interrupt output	

11.3.5.3 PMD Trigger Program Setting Registers 0 to 5

Each of the PMD Trigger Program Registers (ADPSET0 to 5) are comprised of four registers. These four registers are used to select the AD conversion input pin (AIN0 to AIN17). The numbers of these registers correspond to those of the Conversion Result Registers.

Setting the <ENSPnm> to 1 enables the ADPSETnm register. The <AINSPnm> bits are used to select the AIN pin to be used. The numbers of the PMD Trigger Program Setting Registers correspond to those of the Conversion Result Registers. (n= 0 to 5 , m= 0 to 3)

PMD Trigger Program Register 0 (ADPSET0)

ADPSET0
0x4003_0098

	7	6	5	4	3	2	1	0
Bit symbol	ENSP00	-	-	AINSP004	AINSP003	AINSP002	AINSP001	AINSP000
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSP01	-	-	AINSP014	AINSP013	AINSP012	AINSP011	AINSP010
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG1 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSP02	-	-	AINSP024	AINSP023	AINSP022	AINSP021	AINSP020
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSP03	-	-	AINSP034	AINSP033	AINSP032	AINSP031	AINSP030
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG3 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

PMD Trigger Program Register 1 (ADPSET1)

ADPSET1
0x4003_009C

	7	6	5	4	3	2	1	0
Bit symbol	ENSP10	-	-	AINSP104	AINSP103	AINSP102	AINSP101	AINSP100
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSP11	-	-	AINSP114	AINSP113	AINSP112	AINSP111	AINSP110
Read/Write	R/W	R/W		R/W				
After reset	0	0		0 O				
Function	REG1 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSP12	-	-	AINSP124	AINSP123	AINSP122	AINSP121	AINSP120
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSP13	-	-	AINSP134	AINSP133	AINSP132	AINSP131	AINSP130
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG3 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

PMD Trigger Program Register 2(ADPSET2)

ADPSET2
0x4003_00A0

	7	6	5	4	3	2	1	0
Bit symbol	ENSP20	-	-	AINSP204	AINSP203	AINSP202	AINSP201	AINSP200
Read/Write	R/W	R/W		R/W				
After reset	0	0		0 0				
Function	REG0 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSP21	-	-	AINSP214	AINSP213	AINSP212	AINSP211	AINSP210
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG1 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSP22	-	-	AINSP224	AINSP223	AINSP222	AINSP221	AINSP220
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSP23	-	-	AINSP234	AINSP233	AINSP232	AINSP231	AINSP230
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG3 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (01111 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

PMD Trigger Program Register 3(ADPSET3)

ADPSET3
0x4003_00A4

	7	6	5	4	3	2	1	0
Bit symbol	ENSP30	-	-	AINSP304	AINSP303	AINSP302	AINSP301	AINSP300
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENSP31	-	-	AINSP314	AINSP313	AINSP312	AINSP311	AINSP310
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG1 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENSP32	-	-	AINSP324	AINSP323	AINSP322	AINSP321	AINSP320
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG2 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENSP33	-	-	AINSP334	AINSP333	AINSP332	AINSP331	AINSP330
Read/Write	R/W	R/W		R/W				
After reset	0	0		O				
Function	REG3 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

PMD Trigger Program Register 4(ADPSET4)

ADPSET4
0x4003_00A8

	7	6	5	4	3	2	1	0
B t symbol	ENSP40	-	-	AINSP404	AINSP403	AINSP402	AINSP401	AINSP400
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSP41	-	-	AINSP414	AINSP413	AINSP412	AINSP411	AINSP410
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG1 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSP42	-	-	AINSP424	AINSP423	AINSP422	AINSP421	AINSP420
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSP43	-	-	AINSP434	AINSP433	AINSP432	AINSP431	AINSP430
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG3 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

PMD Trigger Program Register 5(ADPSET5)

ADPSET5
0x4003_00AC

	7	6	5	4	3	2	1	0
Bit symbol	ENSP50	-	-	AINSP504	AINSP503	AINSP502	AINSP501	AINSP500
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG0 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSP51	-	-	AINSP514	AINSP513	AINSP512	AINSP511	AINSP510
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG1 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSP52	-	-	AINSP524	AINSP523	AINSP522	AINSP521	AINSP520
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG2 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSP53	-	-	AINSP534	AINSP533	AINSP532	AINSP531	AINSP530
Read/Write	R/W	R/W		R/W				
After reset	0	0		0				
Function	REG3 enable 0: Disable 1: Enable	Reserved		AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

11.3.6 Timer Trigger Program Registers

AD conversion can be started by trigger from Timer5(TMRB5). There are twelve 8-bit registers for programming timer triggers. Setting the <ENSTm> to “1” enables the ADTSETm register. The <AINSTmn> are used to select the AIN pin to be used. The numbers of the Timer Trigger Program Registers correspond to those of the AD Conversion Result Registers.

When finished this AD conversion, interrupt:INTADTMR is generated.

(m= 0 to 11, n= 0 to 4)

Timer Trigger Program Register 03 (ADTSET03)

ADTSET03
0x4003_00B0

	7	6	5	4	3	2	1	0
Bit symbol	ENST0	-	-	AINST04	AINST03	AINST02	AINST01	AINST00
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENST1	-	-	AINST14	AINST13	AINST12	AINST11	AINST10
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENST2	-	-	AINST24	AINST23	AINST22	AINST21	AINST20
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENST3	-	-	AINST34	AINST33	AINST32	AINST31	AINST30
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

Timer Trigger Program Register 47 (ADTSET47)

ADTSET47
0x4003_00B4

	7	6	5	4	3	2	1	0
Bit symbol	ENST4	-	-	AINST44	AINST43	AINST42	AINST41	AINST40
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG4 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	15	14	13	12	11	10	9	8
Bit symbol	ENST5	-	-	AINST54	AINST53	AINST52	AINST51	AINST50
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG5 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	23	22	21	20	19	18	17	16
Bit symbol	ENST6	-	-	AINST64	AINST63	AINST62	AINST61	AINST60
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

	31	30	29	28	27	26	25	24
Bit symbol	ENST7	-	-	AINST74	AINST73	AINST72	AINST71	AINST70
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG7 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

Timer Trigger Program Register 811 (ADTSET811)

ADTSET811
0x4003_00B8

	7	6	5	4	3	2	1	0
Bit symbol	ENST8	-	-	AINST84	AINST83	AINST82	AINST81	AINST80
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENST9	-	-	AINST94	AINST93	AINST92	AINST91	AINST90
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENST10	-	-	AINST104	AINST103	AINST102	AINST101	AINST100
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENST11	-	-	AINST114	AINST113	AINST112	AINST111	AINST110
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

11.3.7 Software Trigger Program Registers

AD conversion can be started by software. There are twelve 8-bit registers for programming software triggers. Setting the <ENSSn> to “1” enables the ADSSETn register. The <AINSSnm> are used to select the AIN pin to be used. The numbers of the Software Trigger Program Registers correspond to those of the Conversion Result Registers.

When finished this AD conversion, interrupt:INTADSFT is generated.

(n= 0 to 11, m= 0 to 4)

Software Trigger Program Register 03 (ADSSET03)

ADSSET03
0x4003_00BC

	7	6	5	4	3	2	1	0
Bit symbol	ENSS0	-	-	AINSS04	AINSS03	AINSS02	AINSS01	AINSS00
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSS1	-	-	AINSS14	AINSS13	AINSS12	AINSS11	AINSS10
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSS2	-	-	AINSS24	AINSS23	AINSS22	AINSS21	AINSS20
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSS3	-	-	AINSS34	AINSS33	AINSS32	AINSS31	AINSS30
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

Software Trigger Program Register 47 (ADSSET47)

ADSSET47
0x4003_00C0

	7	6	5	4	3	2	1	0
Bit symbol	ENSS4	-	-	AINSS44	AINSS43	AINSS42	AINSS41	AINSS40
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG4 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSS5	-	-	AINSS54	AINSS53	AINSS52	AINSS51	AINSS50
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG5 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSS6	-	-	AINSS64	AINSS63	AINSS62	AINSS61	AINSS60
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSS7	-	-	AINSS74	AINSS73	AINSS72	AINSS71	AINSS70
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG7 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

Software Trigger Program Register 811 (ADSSET811)

ADSSET811
0x4003_00C4

	7	6	5	4	3	2	1	0
Bit symbol	ENSS8	-	-	AINSS84	AINSS83	AINSS82	AINSS81	AINSS80
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSS9	-	-	AINSS94	AINSS93	AINSS92	AINSS91	AINSS90
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSS10	-	-	AINSS104	AINSS103	AINSS102	AINSS101	AINSS100
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSS11	-	-	AINSS114	AINSS113	AINSS112	AINSS111	AINSS110
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

11.3.8 Constant Conversion Program Registers

The ADC allow conversion triggers to be constantly enabled. There are twelve 8-bit registers for programming constant triggers. Setting the <ENSAm> to “1” enables the ADASETm register. The <AINSAmn> are used to select the AIN pin to be used. The numbers of the Constant Trigger Program Registers correspond to those of the Conversion Result Registers.

(m= 0 to 11, n= 0 to 4)

Constant Conversion Program Register 03 (ADASET03)

ADASET03
0x4003_00C8

	7	6	5	4	3	2	1	0
Bit symbol	ENSA0	-	-	AINSA04	AINSA03	AINSA02	AINSA01	AINSA00
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG0 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSA1	-	-	AINSA14	AINSA13	AINSA12	AINSA11	AINSA10
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG1 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSA2	-	-	AINSA24	AINSA23	AINSA22	AINSA21	AINSA20
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG2 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSA3	-	-	AINSA34	AINSA33	AINSA32	AINSA31	AINSA30
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG3 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

Constant Conversion Program Register 47 (ADASET47)

ADASET47
0x4003_00CC

	7	6	5	4	3	2	1	0
Bit symbol	ENSA4	-	-	AINSA44	AINSA43	AINSA42	AINSA41	AINSA40
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG4 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSA5	-	-	AINSA54	AINSA53	AINSA52	AINSA51	AINSA50
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG5 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSA6	-	-	AINSA64	AINSA63	AINSA62	AINSA61	AINSA60
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG6 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSA7	-	-	AINSA74	AINSA73	AINSA72	AINSA71	AINSA70
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG7enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

Constant Conversion Program Register 811 (ADASET811)

ADASET811
0x4003_00D0

	7	6	5	4	3	2	1	0
Bit symbol	ENSA8	-	-	AINSA84	AINSA83	AINSA82	AINSA81	AINSA80
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG8 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	15	14	13	12	11	10	9	8
Bit symbol	ENSA9	-	-	AINSA94	AINSA93	AINSA92	AINSA91	AINSA90
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG9 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	23	22	21	20	19	18	17	16
Bit symbol	ENSA10	-	-	AINSA104	AINSA103	AINSA102	AINSA101	AINSA100
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG10 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				
	31	30	29	28	27	26	25	24
Bit symbol	ENSA11	-	-	AINSA114	AINSA113	AINSA112	AINSA111	AINSA110
Read/Write	R/W	R	R	R/W				
After reset	0	0	0	0				
Function	REG11 enable 0: Disable 1: Enable	Always read as 0.	Always read as 0.	AIN select 00000: AIN0 00001: AIN1 : 10001: AIN17 (10010 to 11111: Reserved)				

Note : TMPM382FW/M382FS have 10 analog inputs. The AIN0 to AIN9 are available.

11.4 Operation Descriptions

11.4.1 Analog Reference Voltages

For the High-level and Low-level analog reference voltages, the AVDD5 and AVSS pins are used in ADC. There are no registers for controlling current between AVDD5 and AVSS. Inputs to these pins are fixed.

The ADC can decrease the current consumption when stopping. Setting the <RCUT> to “1” enables the ADMOD3 register.

Note : Analog input pins also use input/output ports(port H/ I/ J), it is recommended for the purpose of maintaining the accuracy of AD conversion result that do not execute input/output instructions during AD conversion. Additionally, do not input widely varying signals into the ports adjacent to analog input pins.

11.4.2 Starting AD Conversion

AD conversion is started by software or one of the following three trigger signals.

- PMD trigger (See “11.3.5 PMD Trigger Program Registers”)
- Timer trigger (TMRB5) (See “11.3.6 Timer Trigger Program Registers”)
- Software trigger (See “11.3.7 Software Trigger Program Registers”)

These start triggers are given priorities as shown below.

PMD trigger 0 > • • • >PMD trigger 3 > Timer trigger > Software trigger > constant trigger

If the PMD trigger occurs while an AD conversion is in progress, the PMD trigger is handled stop the ongoing program and start AD conversion correspond to PMD trigger number.

If a higher-priority trigger occurs while an AD conversion is in progress, the higher-priority trigger is handled after the ongoing program is completed.

It has some delay from generation of trigger to start of AD conversion. The delay depends on the trigger. The following timing chart and table show the delay.

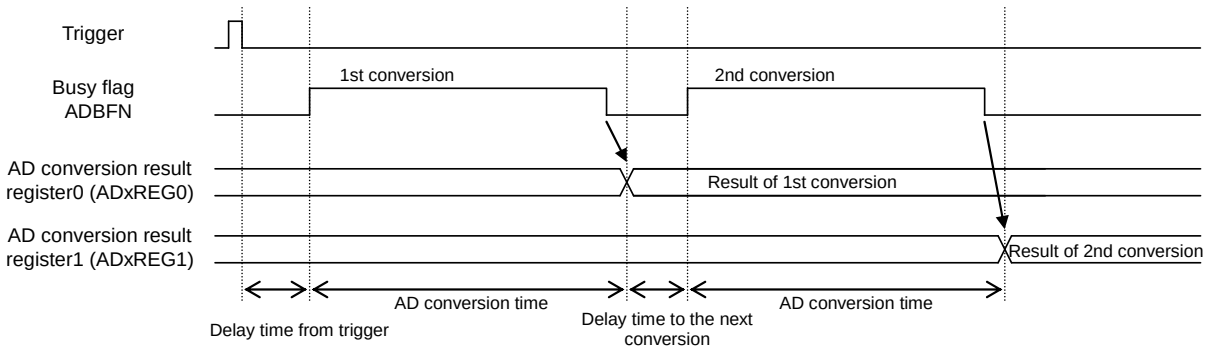


Fig 11.2 Figure Timing chart of A/D conversion

Table 11.2 Table A/D conversion time (SCLK=40MHz)

[μs]

		fsys=40MHz	
	Trigger	MIN	MAX
Delay time from trigger (Note 1)	PMD	0.225	0.3
	TMRB	0.225	0.5
	Software, Constant	0.25	0.525
AD conversion time	−	1.85	
Delay time to the next conversion (Note 2)	PMD	0.175	0.225
	TMRB, Software, Constant	0.175	0.425

Note 1: Delay time from trigger to start of AD conversion.
Note 2: Delay time to the 2nd or after conversion in plural conversions with one trigger.

11.4.3 AD Conversion Monitoring Function

The ADC has the AD conversion monitoring function. When this function is enabled, an interrupt is generated when a conversion result matches the specified comparison value.

To enable the monitoring function set ADCMPCR0<CMP0EN> or ADCMPCR1<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 ADCMCR<ADBIG0> the interrupt (INTADCPA for ADCMPCR0, INTADCPB for ADCMPCR1) 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 (OVRs) is set.

11.5 Timing chart of AD conversion

The following shows a timing chart of software trigger conversion, constant conversion and acceptance of trigger.

11.5.1 Software trigger Conversion

In the software trigger conversion, the interrupt is generated after completion of conversion programmed by ADSSET03, ADSSET47 and ADSSET811.

If the ADMOD1<ADEN> is cleared to “0” during AD conversion, the ongoing conversion stops without storing to the result register.

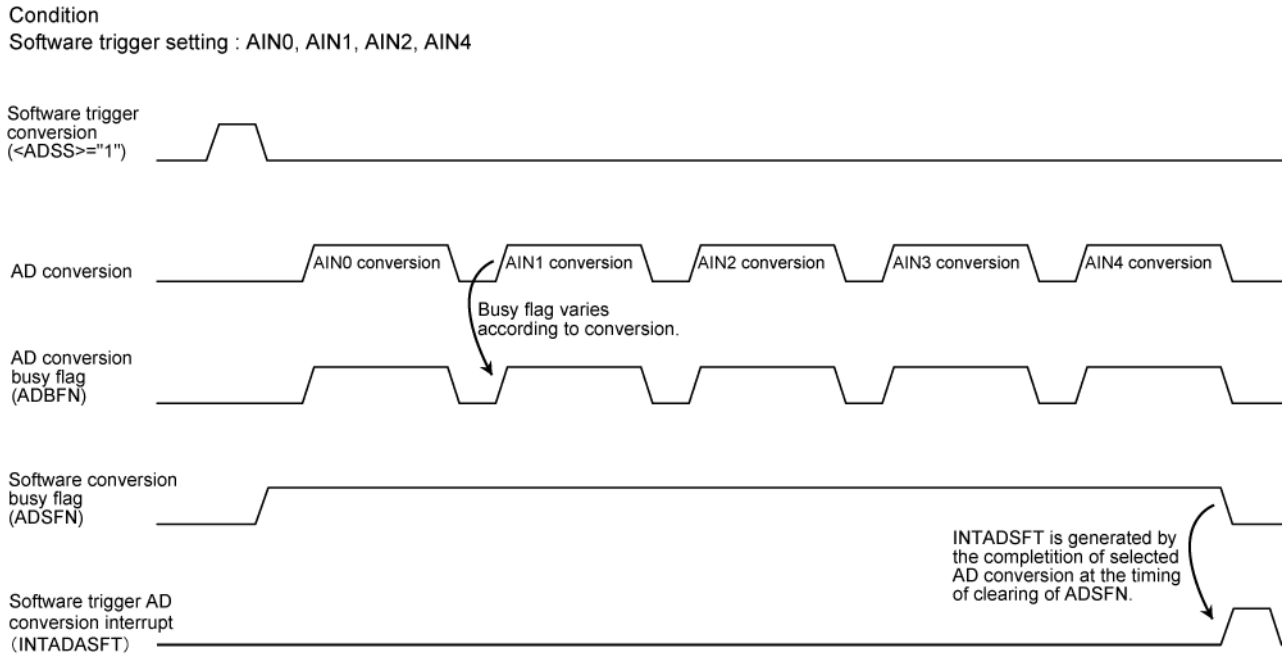


Fig 11.3 Software trigger AD conversion

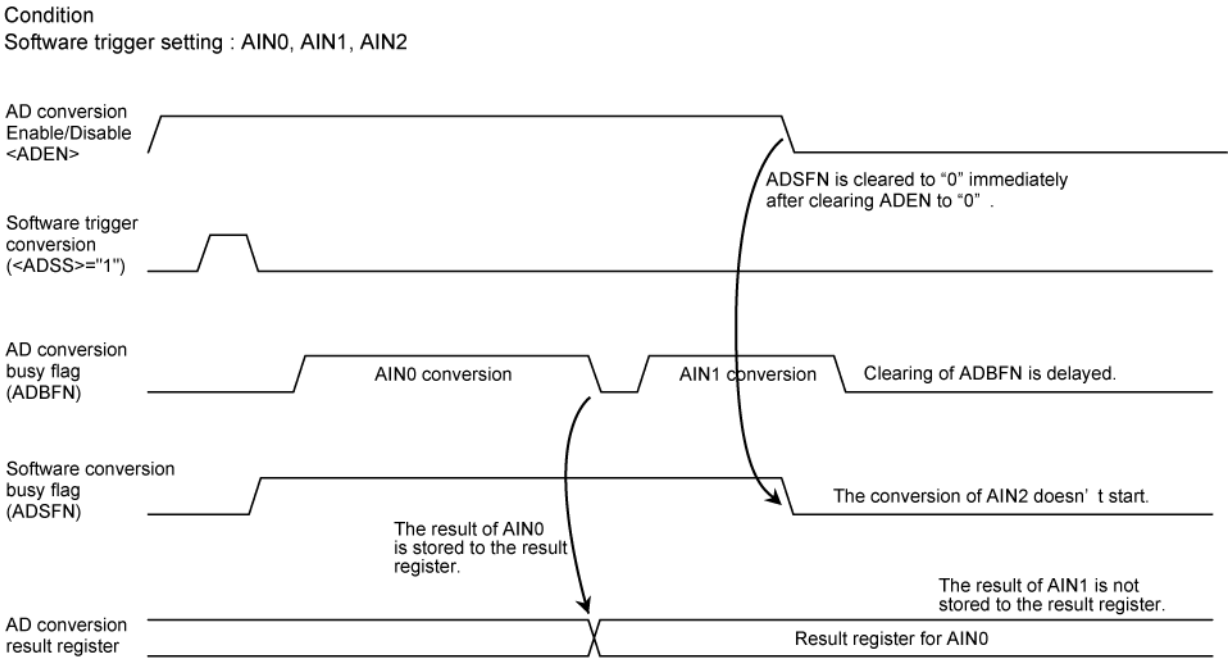


Fig 11.4 Writing “0” to ADEN during the software trigger AD conversion

11.5.2 **Constant Conversion**

In the constant conversion, if the next conversion completes without reading the previous result from the conversion result register, the overrun flag is set to “1”. In this case, the previous conversion result in the conversion result register is overwritten by the next result. The overrun flag is cleared by the conversion result.

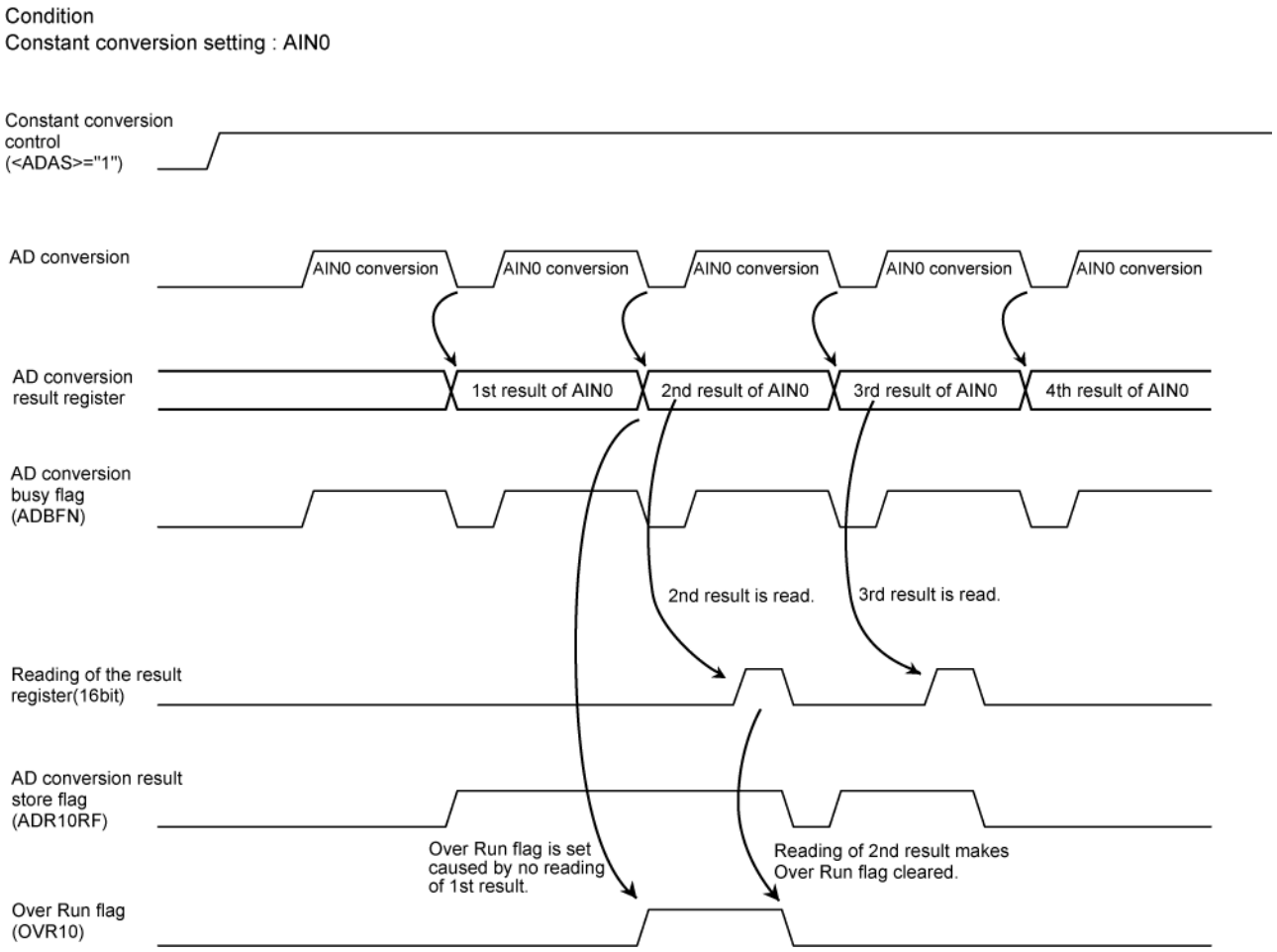


Fig 11.5 Constant conversion

11.5.3 AD conversion by trigger

If the PMD trigger is occurred during the software trigger conversion, the ongoing conversion stops immediately.

If the timer trigger is occurred during the software trigger conversion, the ongoing conversion stops after the completion of ongoing conversion. After the completion of conversion by trigger, the software trigger conversion starts from the beginning programmed by ADSSET03, ADSSET47 and ADSSET811.

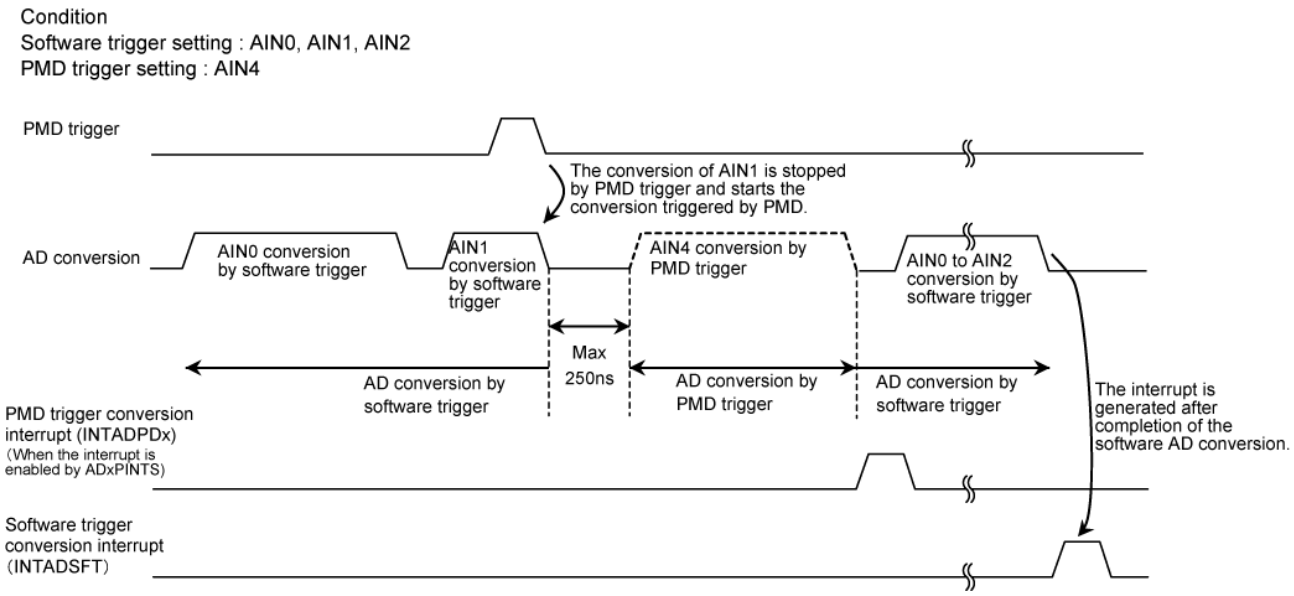


Fig 11.6 AD conversion by PMD trigger

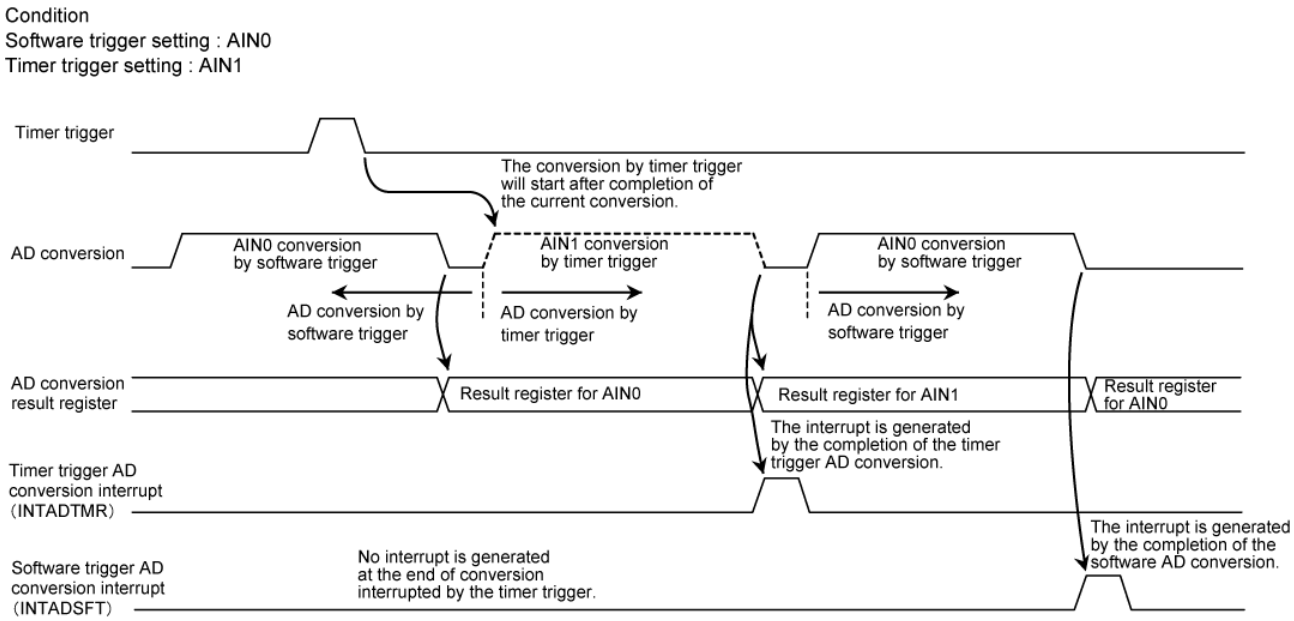


Fig 11.7 AD conversion by timer trigger (1)

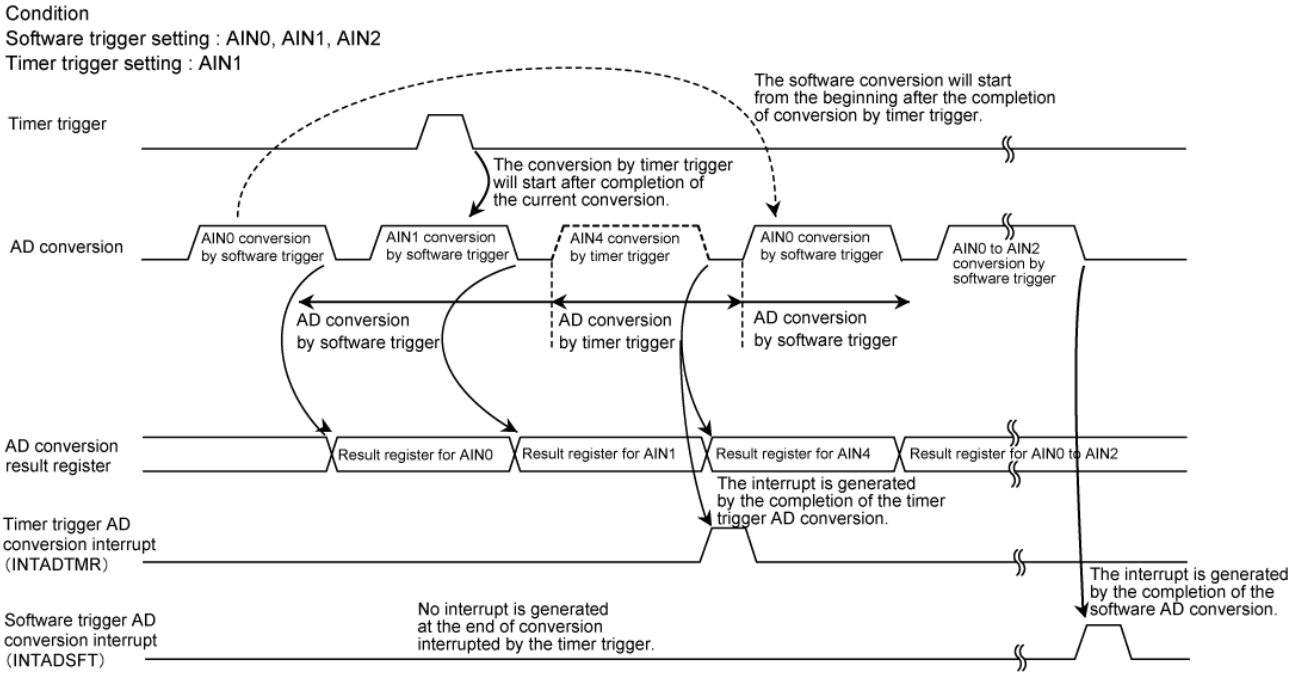


Fig 11.8 AD conversion by timer trigger (2)

Cautions
The result value of AD conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. When using analog input pins and ports alternately, do not read and write ports during conversion because the conversion accuracy may be reduced. Also the conversion accuracy may be reduced if the output ports current fluctuate during AD conversion. Please take counteractive measures with the program such as averaging the AD conversion results.

12 Encoder Input Circuit (ENC)

Important
TMPM382 (64-pin version) does not implement ENC0 and ENC1.
Please do not use these functions.

The TMPM380 has a two-channel incremental encoder interface (ENC0/1), which can determine the direction and the absolute position of a motor, based on input signals from an incremental encoder.

The discussions in this chapter apply to ENC0. For ENC1, the names of registers, interrupt signals and pins in the following text should be replaced as appropriate, as shown in Table 12-1 to Table 12-3.

Table 12-1 List of the ENC Registers

Register	ENC0		ENC1	
	Register Symbol	Address	Register Symbol	Address
Encoder Input Control Register	EN0TNCR	0x4001_0400	EN1TNCR	0x4001_0500
Encoder Counter Reload Register	EN0RELOAD	0x4001_0404	EN1RELOAD	0x4001_0504
Encoder Compare Register	EN0INT	0x4001_0408	EN1INT	0x4001_0508
Encoder Counter	EN0CNT	0x4001_040C	EN1CNT	0x4001_050C

Table 12-2 Interrupt Sources

Interrupt Source	ENC0	ENC1
ENC interrupt	INTENC0	INTENC1

Table 12-3 Pin Names

Pin	ENC0	ENC1
Channel A input pin	PD0/ENCA0	PF2/ENCA1
Channel B input pin	PD1/ENCB0	PF3/ENCB1
Channel Z input pin	PD2/ENCZ0	PF4/ENCZ1

12.1 Outline

The ENC can be configured to operate in one of four different modes: Encoder mode, two Sensor modes (Event count mode, Timer count mode) and Timer mode. And it also has some functions as below.

- Supports incremental encoders and Hall sensor ICs. (signals of Hall sensor IC can be input directly)
- 24-bit general-purpose timer mode
- Multiply-by-4 (multiply-by-6) logic
- Direction discriminator
- 24-bit counter
- Comparator enable/disable
- Interrupt request output
- Digital noise filters for input signals

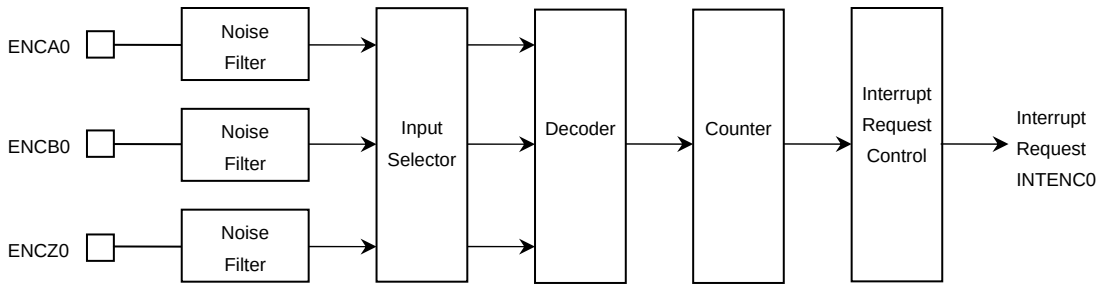


Fig 12-1 ENC Block Diagram

12.1.1 Encoder Mode

In Encoder mode, the ENC provides high-speed position tracking, based on the A/B or A/B/Z input signals from an incremental encoder.

- Event (rotation pulse) sensing: Programmable to generate an interrupt on each event.
- Event counter: Programmable to generate an interrupt at a preset count (for positional displacement calculation).
- Direction discrimination
- Up/down counting (dynamically selectable)
- Programmable counter period

12.1.2 Sensor Modes

In Sensor modes, the ENC provides low-speed position (zero-cross) tracking, based on either the A(U)/B(V) or A(U)/B(V)/Z(W) input signals from a Hall sensor.

There are two operating modes: Event Count mode and Timer Count mode (which runs with fsys).

12.1.2.1 Event Count Mode

- Event (rotation pulse) sensing: Programmable to generate an interrupt on each event.
- Event counter: Programmable to generate an interrupt at a preset count (for positional displacement calculation).
- Direction discrimination

12.1.2.2 Timer Count Mode

- Event (rotation pulse) sensing: Programmable to generate an interrupt on each event.
- Timer counting operation
- Direction discrimination
- Input capture functions

  Event capture (event interval measurement): Programmable to generate an interrupt.

  Software capture
- Event timeout error (timer compare): Programmable to generate an interrupt at a preset count.
- Revolution error: Error flag that indicates a change of the rotation direction

12.1.3 Timer Mode

The ENC can serve as a 24-bit general-purpose timer.

- 24-bit up-counter
- Counter clear: via a software clear bit, or at a preset count, or by an external trigger input, or on overflow of the free-running counter.
- Timer compare: Programmable to generate an interrupt at a preset count.
- Input capture functions

  External trigger capture: Programmable to generate an interrupt.

  Software capture

12.2 Control Registers

Encoder 0 Input Control Register

EN0TNCR (0x4001_0400)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0
Description	Reading these bits returns a 0.							

	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	MODE1	MODE0	P3EN
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	Reading these bits returns a 0.					ENC Operating Mode 00: Encoder mode 01: Sensor Event Count mode 10: Sensor Timer Count mode 11: Timer mode		[In Sensor mode] 2/3-Phase Input Select 0: 2-phase 1: 3-phase

	15	14	13	12	11	10	9	8
Bit Symbol	CMP	REVERR	UD	ZDET	SFTCAP	ENCLR	ZESEL	CMPEN
Read/Write	R	R	R	R	W	W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	Compare Flag 0: ― 1: Counter compared This bit is cleared on a read.	[In Sensor Timer Count mode] Revolution Error 0: ― 1: Error occurred. This bit is cleared on a read.	Rotation Direction 0: CCW 1: CW	Z_ Detected 0: Not detected 1: Z phase detected	[In Sensor Timer Count and Timer modes] Software Capture 0: ― 1: Software capture	Encoder Counter Clear 0: ― 1: Clears the counter.	[In Timer mode] Z Trigger Edge Select 0: Rising edge 1: Falling edge	Compare Enable 0: Compare disabled 1: Compare enabled

	7	6	5	4	3	2	1	0
Bit Symbol	ZEN	ENRUN	NR1	NR0	INTEN	ENDEV2	ENDEV1	ENDEV0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	Z Phase Enable 0: Disabled 1: Enabled	ENC Run 0: Disabled 1: Enabled	Noise Filter 00: No filtering 01: Filters out pulses narrower than 31/fsys (387.5 ns@80 MHz). 10: Filters out pulses narrower than 63/fsys(787.5 ns@80 MHz). 11: Filters out pulses narrower than 127/fsys(1587 ns@80 MHz).		ENC Interrupt Enable 0: Disabled 1: Enabled	Encoder Pulse Division Factor 000: ÷1 001: ÷2 010: ÷4 011: ÷8 100: ÷16 101: ÷32 110: ÷64 111: ÷128		

Description:

<MODE1:0>: ENC Operating Mode

00: Encoder mode

01: Sensor Event Count mode

10: Sensor Timer Count mode

11: Timer mode

<MODE1:0> selects an operating mode for the ENC.

Operating modes are defined by <MODE1:0>, <P3EN> and <ZEN> as shown in the following table. There are a total of eight operating modes.

<MODE1>	<MODE0>	<ZEN>	<P3EN>	Input signal Used	Operating Mode
0	0	0	0	A, B	Encoder mode
		1		A, B, Z	Encoder mode (using Z)
0	1	0	0	A,B	Sensor Event Count mode (2 phase inputs)
			1	A, B, Z	Sensor Event Count mode (3 phase inputs)
1	0	0	0	A, B	Sensor Timer Count mode (2 phase inputs)
			1	A, B, Z	Sensor Timer Count mode (3 phase inputs)
1	1	0	0	-	Timer mode
		1		Z	Timer mode (using Z)

<P3EN>: 2/3-Phase Input Select

0: 2-phase

1: 3-phase

<P3EN> selects the number of phase input pins used.
If <P3EN> is cleared to 0, the ENC decodes two phase inputs.
If <P3EN> is set to 1, the ENC decodes three phase inputs.
In Timer mode, this bit has no effect.

Note: <P3EN> must always be cleared in Encoder mode, irrespective of the number of phase input pins used.

<CMP>: Compare Flag

0: －

1: Counter compared

<CMP> is set to 1 when the counter value has been compared to the value programmed in the EN0INT register.
<CMP> is cleared to 0 on a read. <CMP> remains cleared when <ENRUN> = 0. Writing to <CMP> has no effect.

<REVERR>: Revolution Error

0: －

1: Error occurred.

In Sensor Timer Count mode, <REVERR> is set to 1 when a change in the rotation direction has been detected. <REVERR> is cleared to 0 on a read. <REVERR> remains cleared when <ENRUN> = 0. Writing to <REVERR> has no effect.
This bit has an effect only in Sensor Timer Count mode.

Note: Once software has changed the operating mode of the ENC, <REVERR> must be cleared by reading it.

<U/D>: Rotation Direction

0: Counterclockwise (CCW)

1: Clockwise (CW)

The quadrature signals A and B identify the motor rotation direction. <U/D> is set to 1 when the CW direction is indicated (signal A of the incremental encoder signal is ahead of signal B). <U/D> is cleared to 0 when the CCW direction is indicated (signal A is behind signal B). <U/D> remains cleared while <ENRUN> = 0.

<ZDET>: Z Detected
0: Not detected
1: Detected

<ZDET> is set to 1 on the first edge of Z input signal (ENCZ) after <ENRUN> is written from 0 to 1. This occurs on a rising edge of the signal Z during CW rotation or on a falling edge of Z during CCW rotation.
<ZDET> remains cleared while <ENRUN> = 0.
<ZEN> has no influence on the value of <ZDET>.
<ZDET> remains cleared in Sensor Event Count and Sensor Timer Count modes.

<SFTCAP>: Software Capture
0: －
1: Software capture

If <SFTCAP> is set to 1, the value of the encoder counter is captured into the EN0CNT register.
Writing a 0 to <SFTCAP> has no effect. Reading <SFTCAP> always returns a 0.
In Encoder and Sensor Event Count modes, <SFTCAP> has no effect; a write of a 1 to this bit is ignored.

<ENCLR>: Encoder Counter Clear
0: －
1: Clears the encoder counter.

Writing a 1 to <ENCLR> clears the encoder counter to 0. Once cleared, the encoder counter restarts counting from 0. Writing a 0 to <ENCLR> has no effect. Reading <ENCLR> always returns a 0.

<ZESEL>: Z Trigger Edge Select
0: Uses a rising edge of the ENCZ as an external trigger input.
1: Uses a falling edge of the ENCZ as an external trigger input.

<ZESEL> selects the edge of the ENCZ that should be used as an external trigger in Timer mode.
In the other operating modes, <ZESEL> has no effect.

<CMPEN>: Compare Enable
0: Compare disabled
1: Compare enabled

If <CMPEN> is set to 1, the value of the encoder counter is compared to the value programmed in the EN0INT register. If <CMPEN> is cleared to 0, this comparison is not done.

<ZEN>: Z Phase Enable
0: Disabled
1: Enabled

- In Encoder mode
  <ZEN> controls whether to clear the encoder counter (EN0CNT) on the rising or falling edge of Z.
  When <ZEN> = 1, the encoder counter is cleared on the rising edge of the ENCZ input if the motor is rotating in the CW direction; the encoder counter is cleared on the falling edge of ENCZ if the motor is rotating in the CCW direction. If the edges of ENCLK (multiply_by_4 clock derived from the decoded A and B signals) and the edge of ENCZ coincide, the encoder counter is cleared to 0 without incrementing or decrementing (i.e., the clear takes precedence).

- In Timer mode
  <ZEN> controls whether to use the ENCZ signal as an external trigger input.
  When <ZEN> = 1, the value of the encoder counter is captured into the EN0INT register and cleared to 0 on the edge of ENCZ selected by <ZESEL>.

In the other operating modes, <ZEN> has no effect.

<ENRUN>: ENC Run
0: Disabled
1: Enabled

Setting <ENRUN> to 1 and clearing <ZDET> to 0 enables the encoder operation.
Clearing <ENRUN> to 0 disables the encoder operation.

There are counters and flags that are and are not cleared even if the <ENRUN> bit is cleared to 0.

The following table shows the states of the counters and flags, depending on the value of <ENRUN>.

Internal Counter / Flag	When <ENRUN> = 0 (After reset)	When <ENRUN> = 1 (During active operation)	When <ENRUN> = 0 (During idle mode)	How to clear a counter or flag when <ENRUN> = 0
Encoder counter	0x000000	Counting	Keeps the current value.	Software clear (Write a 1 to <ENCLR>.)
Noise filter counter	0y0000000	Counting up	Counting up (Continues with noise filtering.)	Cleared only by reset.
Encoder pulse division counter	0x00	Counting down	Stopped and cleared	Cleared when <ENRUN> = 0.
Compare flag <CMP>	0	Set to 1 upon comparison; cleared to 0 on a read.	Cleared	Cleared when <ENRUN> = 0.
Revolution Error flag <REVERR>	0	Set to 1 upon an error; cleared to 0 on a read.	Cleared	Cleared when <ENRUN> = 0.
Z Detected flag <ZDET>	0	Set to 1 on detection of Z.	Cleared	Cleared when <ENRUN> = 0.
Rotation Direction Bit <U/D>	0	Set or cleared according to rotation direction.	Cleared	Cleared when <ENRUN> = 0.

<NR1:0>: Noise Filter
00: No filtering
01: Filters out pulses narrower than 31/fsys as noises.
10: Filters out pulses narrower than 63/fsys as noises.
11: Filters out pulses narrower than 127/fsys as noises.

The digital noise filters remove pulses narrower than the width selected by <NR1:0>.

<INTEN>: ENC Interrupt Enable
0: Disabled
1: Enabled

<INTEN> enables and disables the ENC interrupt.
Setting <INTEN> to 1 enables interrupt generation. Clearing <INTEN> to 0 disables interrupt generation.

<ENDEV2:0>: Encoder Pulse Division Factor

000: ÷1
001: ÷2
010: ÷4
011: ÷8
100: ÷16
101: ÷32
110: ÷64
111: ÷128

The frequency of the encoder pulse is divided by the factor specified by <ENDEV2:0>.
The divided signal determines the interval of the event interrupt.

Encoder 0 Counter Reload Register

EN0RELOAD (0x4001_0404)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0
Description	Reading these bits returns a 0.							

	23	22	21	20	19	18	17	16
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0
Description	Reading these bits returns a 0.							

	15	14	13	12	11	10	9	8
Bit Symbol	RELOAD15	RELOAD14	RELOAD13	RELOAD12	RELOAD11	RELOAD10	RELOAD9	RELOAD8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	(See the description below.)							

	7	6	5	4	3	2	1	0
Bit Symbol	RELOAD7	RELOAD6	RELOAD5	RELOAD4	RELOAD3	RELOAD2	RELOAD1	RELOAD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	Setting the Encoder Counter Period (Multiplied by 4 (or 6)) 0x0000 thru. 0xFFFF When Z is used: Specifies the number of counter pulses per revolution. When Z is not used: Specifies the number of counter pulses per revolution minus 1.							

Description:

<RELOAD15:0>: setting the Encoder Counter Period

- In Encoder mode
  <RELOAD15:0> defines the encoder counter period multiplied by 4.
  If the encoder counter is configured as an up-counter, it increments up to the value programmed in <RELOAD15:0> and then wraps around to 0 on the next ENCLK. If the encoder counter is configured as a down-counter, it decrements to 0 and then is reloaded with the value of <RELOAD15:0> on the next ENCLK.

  The EN0RELOAD register is only used in Encoder mode.

Encoder 0 Compare Register

EN0INT (0x4001_0408)

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0
Description	Reading these bits returns a 0.							

	23	22	21	20	19	18	17	16
Bit Symbol	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	See the description below. <INT23:16> are used only in Sensor Timer Count mode and Timer mode							

	15	14	13	12	11	10	9	8
Bit Symbol	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	See the description below.							

	7	6	5	4	3	2	1	0
Bit Symbol	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	In Encoder mode: Generates an interrupt at the programmed encoder pulse count (0x0000 thru. 0xFFFF). In Sensor Event Count mode: Generates an interrupt at the programmed encoder pulse count (0x0000 thru. 0xFFFF). In Sensor Timer Count mode: Generates an interrupt when the counter has reached the programmed value without detecting a pulse (0x000000 thru. 0xFFFFFF). In Timer mode: Generates an interrupt when the counter value has reached the programmed value (0x000000 thru. 0xFFFFFF).							

Description:

<INT15:0>: Counter Compare Value

- In Encoder mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT15:0>, provided <CMPEN> is set to 1. At this time, the event counter interrupt (INTENC) is asserted if <INTEN> is set to 1.
  However, when <ZEN> = 1, INTENC is not asserted until <ZDET> is set to 1. In Encoder mode, <INT23:16> are not used (and are ignored even if programmed).

- In Sensor Event Count mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT15:0>, provided <CMPEN> is set to 1. At this time, the interrupt request (INTENC) is asserted if <INTEN> is set to 1. The value of <ZEN> has no effect on this interrupt generation.
  In Sensor Event Count mode, <INT23:16> are not used (and are ignored even if programmed).

- In Sensor Timer Count mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT23:0>, provided <CMPEN> is set to 1. This indicates the absence of a pulse for an abnormally long period. At this time, the interrupt request (INTENC) is asserted if <INTEN> is set to 1. The value of <ZEN> has no effect on this interrupt generation.

- In Timer mode
  <CMP> is set to 1 when the value of the encoder counter has reached the value of <INT23:0>, provided <CMPEN> is set to 1. At this time, the timer compare interrupt (INTENC) is asserted if <INTEN> is set to 1. The value of <ZEN> has no effect on this interrupt generation.

Encoder 0 Counter Register

EN0CNT (0x4001_040C

	31	30	29	28	27	26	25	24
Bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0
Description	Reading these bits returns a 0.							

	23	22	21	20	19	18	17	16
Bit Symbol	CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	See the description below. <CNT23:16> are used only in Sensor Timer Count mode and Timer mode. In Encoder mode and Sensor Event Count mode, reading these bits returns a 0.							

	15	14	13	12	11	10	9	8
Bit Symbol	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	See the description below.							

	7	6	5	4	3	2	1	0
Bit Symbol	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Description	In Encoder mode: Number of encoder pulses 0x0000 thru. 0xFFFF In Sensor mode: Pulse detection time or the encoder counter value captured under software control 0x000000 thru. 0xFFFFFF In Timer mode: Iencoder counter value captured by hardware signaling or under software control 0x000000 thru. 0xFFFFFF							

Description:

<CNT15:0>: Encoder Counter/Captured Value

- In Encoder mode
  The value of encoder count.can be read out from <CNT15:0>.
  In Encoder mode, the encoder counter counts up or down on each encoder pulse (ENCLK).
  During CW rotation, encoder counter counts up; when it has reached the value of <RELOAD15:0>, it wraps around to 0 on the next ENCLK.
  During CCW rotation, encoder counter counts down; when it has reached 0, it is reloaded with the value of <RELOAD15:0> on the next ENCLK.

- In Sensor Event Count mode
  The value of encoder count.can be read out from <CNT15:0>.
  In Sensor Event Count mode, the encoder counter counts up or down on each encoder pulse (ENCLK).
  During CW rotation, encoder counter counts up; when it has reached0xFFFF, it wraps around to 0 on the next ENCLK.
  During CCW rotation, encoder counter counts down; when it has reached 0, it wraps around to 0xFFFF on the next ENCLK.

- In Sensor Timer Count mode
  <CNT23:0> contains the value of the encoder counter captured by either the encoder pulse (ENCLK) or software setting <SFTCAP> to 1. The captured value in <CNT23:0> is cleared to 0 on system reset. It can also be cleared by clearing the counter via setting <ENCLR> to 1 and then setting <SFTCAP> to 1.
  In Sensor Timer Count mode, the encoder counter is configured as a free-running counter that counts up with fsys. The encoder counter is cleared to 0 when the encoder pulse (ENCLK) is detected. When it has reached 0xFFFFFF, it wraps around to 0 automatically.

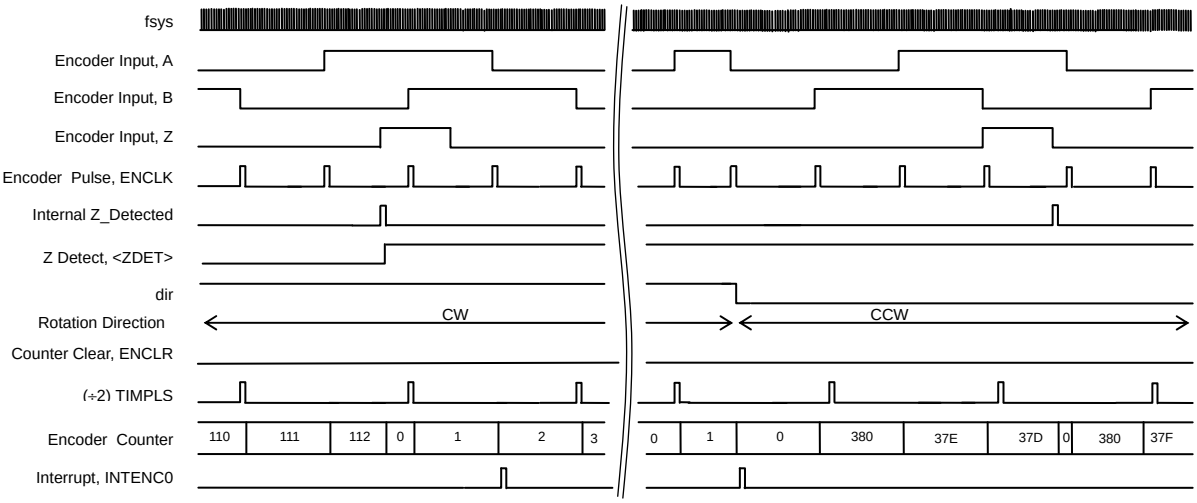
- In Timer mode
  <CNT23:0> contains the value of the encoder counter captured by software setting <SFTCAP> to 1. When <ZEN> = 1, the value of the encoder counter is also captured into <CNT23:0> on the Z ENCZ edge selected by <ZESEL>.
  In Timer mode, the encoder counter is configured as a free-running counter that counts up with fsys. When it has reached 0xFFFFFF, it wraps around to 0 automatically.

12.3 Functional Description

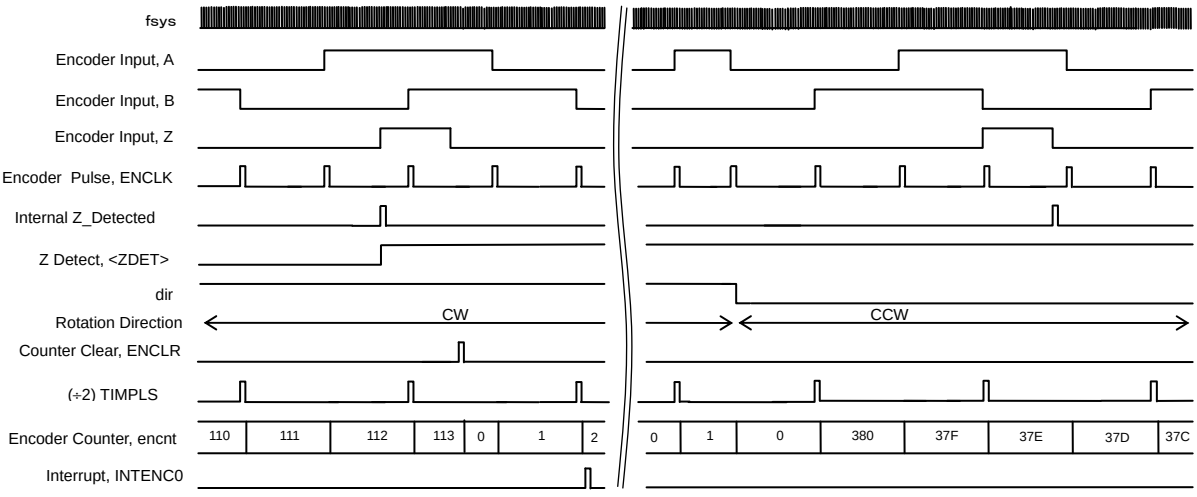
12.3.1 Operating Modes

12.3.1.1 Encoder Mode

(1) When <ZEN> = 1 (<RELOAD> = 0x0380, <EN0INT> = 0x0002)



(2) When <ZEN> = 0 (<RELOAD> = 0x0380, <EN0INT> = 0x0002)



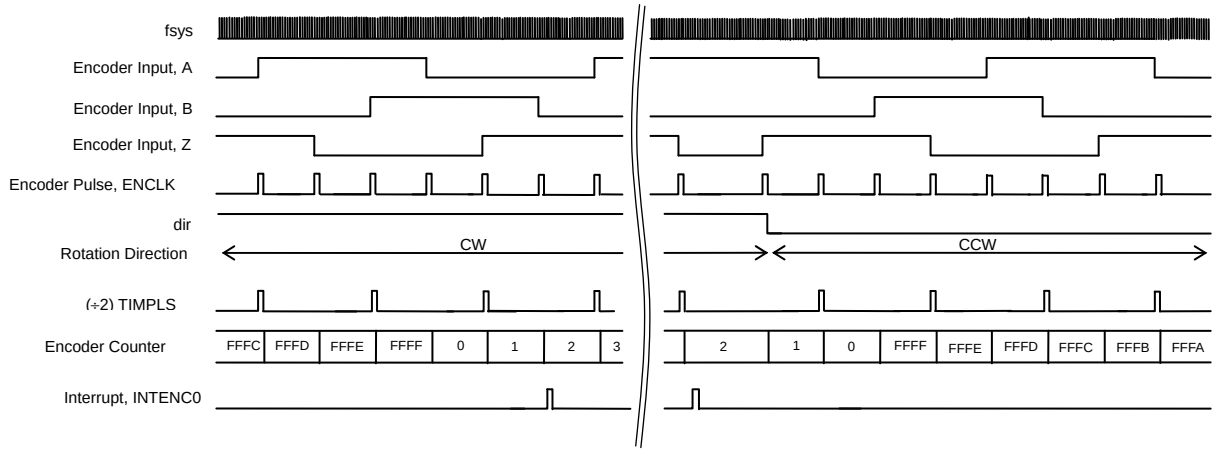
- In Encoder mode, the incremental encoder inputs of the TMPM370 should be connected to the A, B and Z channels. The encoder counter counts pulses of ENCLK, which is multiplied_by_4 clock derived from the decoded A and B quadrature signals.
- During CW rotation (i.e., when A leads B), the encoder counter counts up; when it has reached the value of <RELOAD>, it wraps around to 0 on the next ENCLK.
- During CCW rotation (i.e., when A lags B), the encoder counter counts down; when it has reached 0x0000, it is reloaded with the value of <RELOAD> on the next ENCLK.
- Additionally, when <ZEN> = 1, the encoder counter is cleared to 0 on the rising edge of Z during CW rotation and on the falling edge of Z during CCW rotation (at the internal Z_Detected timing).

If the ENCLK and Z edges coincide, the encoder counter is cleared to 0 without incrementing or decrementing.

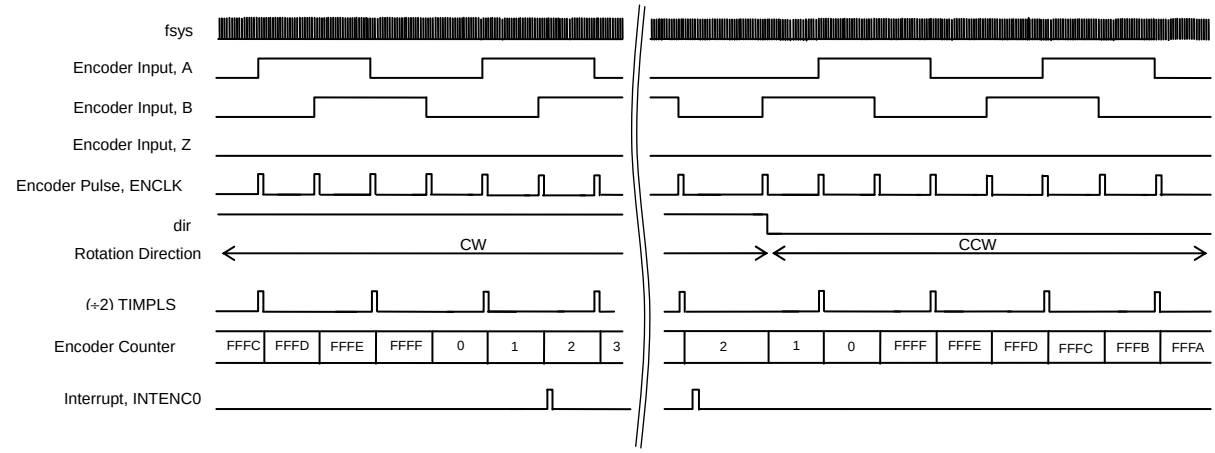
- When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.
- <U/D> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <EN0INT>. When <ZEN> = 1, however, an interrupt does not occur while <ZDET> = 0.
- Clearing <ENRUN> to 0 clears <ZDET> and <U/D> to 0.

12.3.1.2 Sensor Event Count Mode

(1) When <P3EN> = 1 (<EN0INT> = 0x0002)



(2) When <P3EN> = 0 (<EN0INT> = 0x0002)

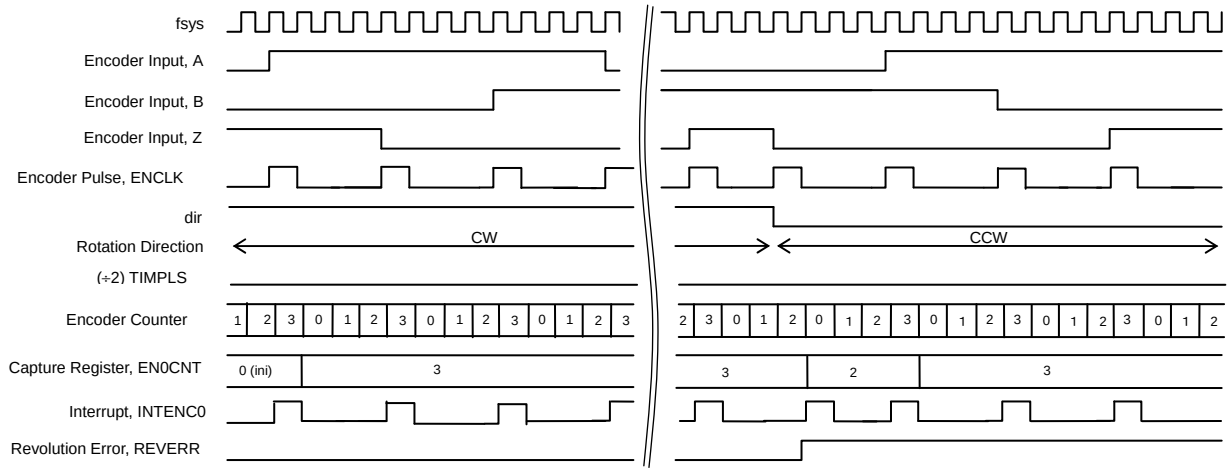


- In Sensor Event Count mode, the Hall sensor inputs of the TMPM370 should be connected to the A(U), B(V) and Z(W) channels. The encoder counter counts the pulses of ENCLK, which is either multiplied_by_4 clock (when <P3EN> = 0) derived from the decoded A(U) and B(V) signals or multiplied_by_6 clock (when <P3EN> = 1) derived from the decoded A(U), B(V) and Z(W) signals.
- During CW rotation, the encoder counter counts up; when it has reached 0xFFFF, it wraps around to 0 on the next ENCLK.

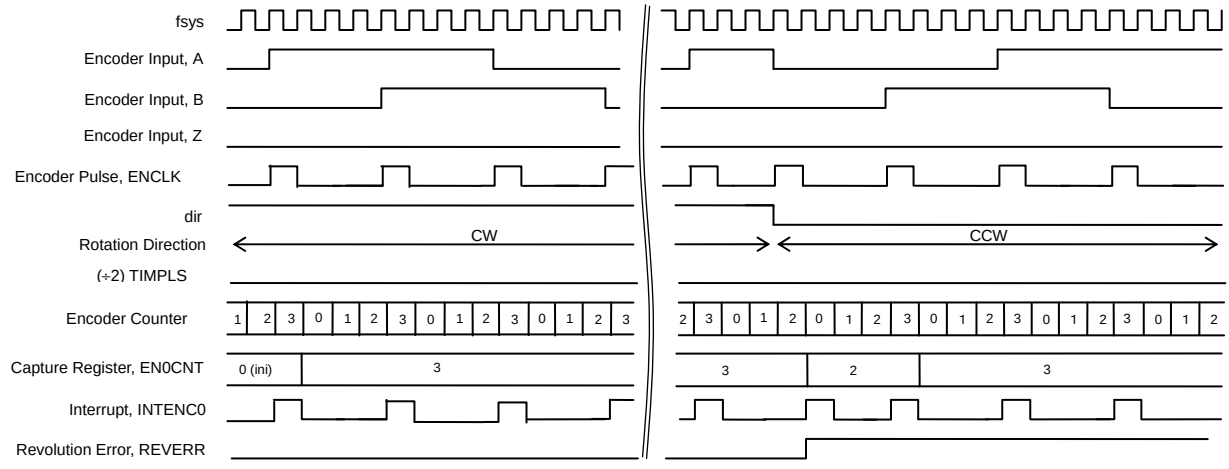
- During CCW rotation, the encoder counter counts down; when it has reached 0x0000, it wraps around to 0xFFFF on the next ENCLK.
- When <ENCLR> is set to 1, causing the internal counter to be cleared to 0.
  - <U/D> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
  - TIMPLS, which is derived by dividing ENCLK by a programmed factor, can be driven out externally.
  - If <CMPEN> is set to 1, an interrupt is generated when the value of the internal counter has reached the value of <EN0INT>.
  - Clearing <ENRUN> to 0 clears <U/D> to 0.

12.3.1.3 Sensor Timer Count Mode

(1) When <P3EN> = 1 (<EN0INT> = 0x0002)



(2) When <P3EN> = 0 (<EN0INT> = 0x0002)



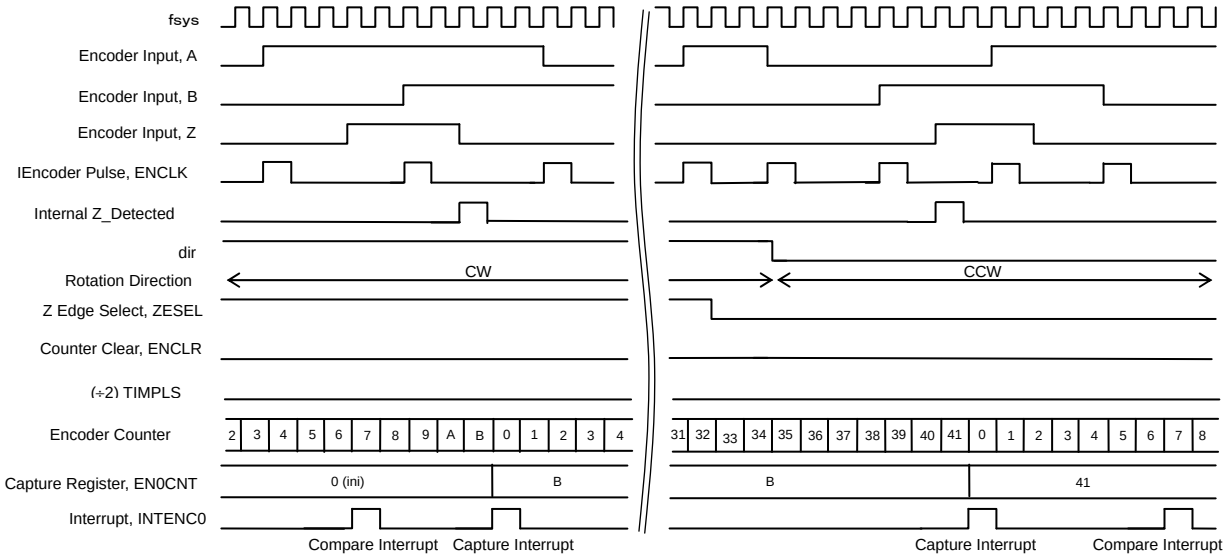
- In Sensor Timer Count mode, the Hall sensor inputs of the TMPM370 should be connected to the A(U), B(V) and Z(W) channels. The encoder counter measures the interval between two contiguous pulses of ENCLK, which is either multiplied_by_4 clock (when <P3EN> = 0) derived from the decoded A(U) and B(V) signals or multiplied_by_6 clock (when <P3EN> = 1) derived from the decoded A(U), B(V) and Z(W) signals.
- The encoder counter always counts up; it is cleared to 0 on ENCLK. When the encoder counter

has reached 0xFFFFFF, it wraps around to 0.

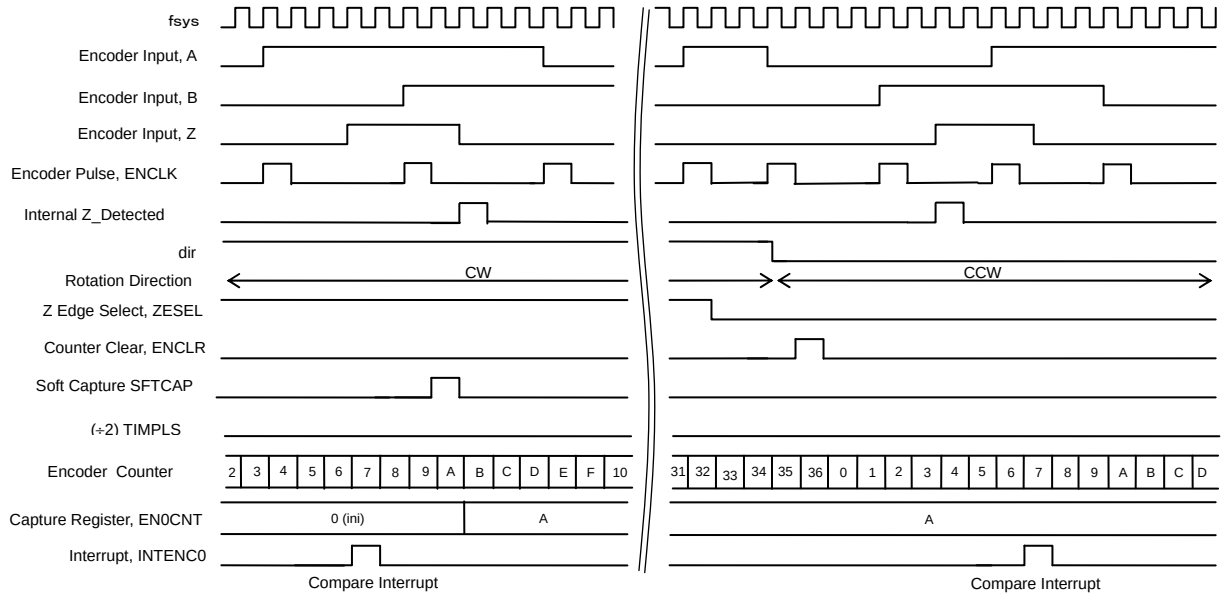
- When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.
- ENCLK causes the value of the encoder counter to be captured into the EN0CNT register. The captured counter value can be read out of EN0CNT.
- Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.
- Clearing <ENRUN> to 0 clears <U/D> to 0.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <EN0INT>.
- <U/D> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- <REVERR> is set to 1 when the rotation direction has changed. This bit is cleared to 0 on a read.
- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

12.3.1.4 Timer Mode

(1) When <ZEN> = 1 (<EN0INT> = 0x0006)



(2) When <ZEN> = 0 (<EN0INT> = 0x0006)



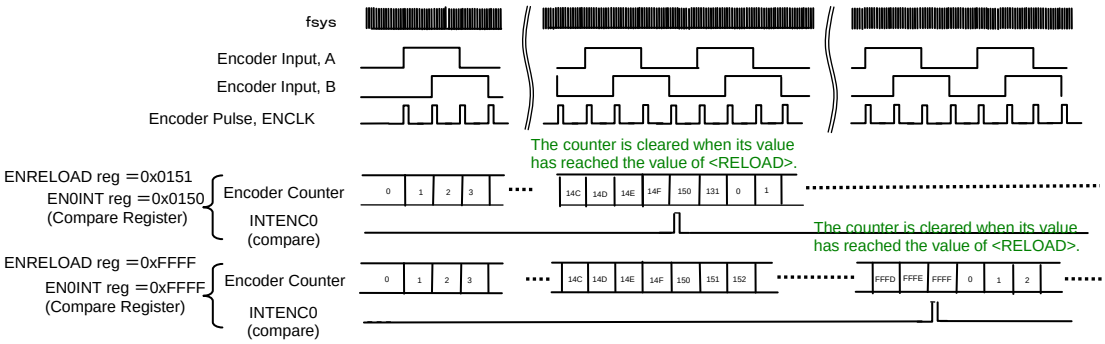
- When <ZEN> = 1, the Z input pin is used as an external trigger. When <ZEN> = 0, no external input is used to trigger the timer.
- The encoder counter always counts up. If <ZEN> = 1, the counter is cleared to 0 on the selected edge of Z (at the internal Z_Detected timing): a rising edge when <ZESEL> = 0 and a falling edge when <ZESEL> = 1. When the encoder counter has reached 0xFFFFFF, it wraps around to 0.
- When <ENCLR> is set to 1, causing the encoder counter to be cleared to 0.
- Z_Detected causes the value of the encoder counter to be captured into the ENCNT register. The captured counter value can be read out of ENCNT.
- Setting the software capture bit, <SFTCAP>, to 1 causes the value of the encoder counter to be

captured into the ENCNT register. This capture operation can be performed at any time. The captured counter value can be read out of ENCNT.

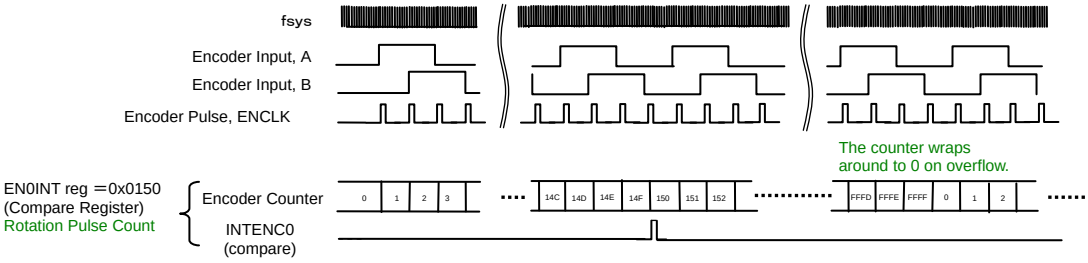
- <U/D> is set to 1 during CW rotation and cleared to 0 during CCW rotation.
- If <CMPEN> is set to 1, an interrupt is generated when the value of the encoder counter has reached the value of <ENINT>.
- Clearing <ENRUN> to 0 clears <U/D> to 0.
- The value of the ENCNT register (the captured value) is retained, regardless of the value of <ENRUN>. The ENCNT register is only cleared by a reset.

12.3.2 Counter Operation and Interrupt Generation When <CMPEN> = 1

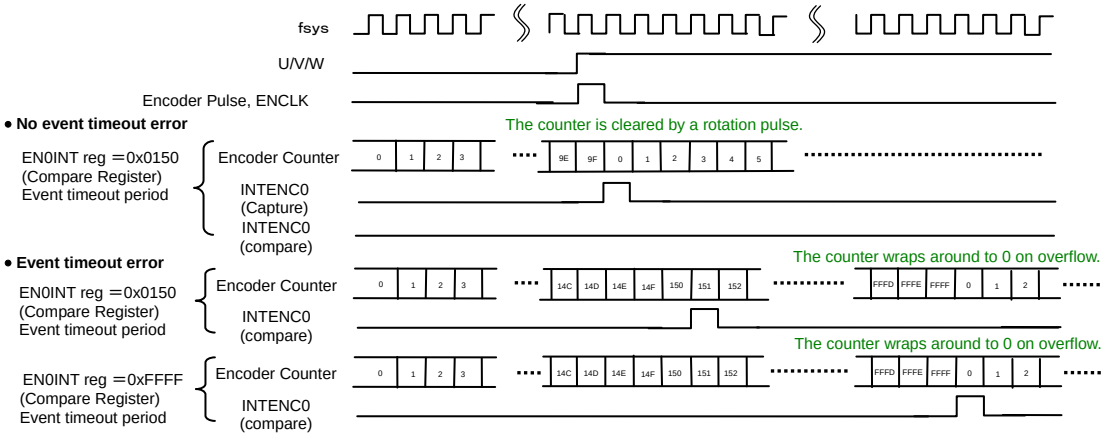
12.3.2.1 Encoder Mode



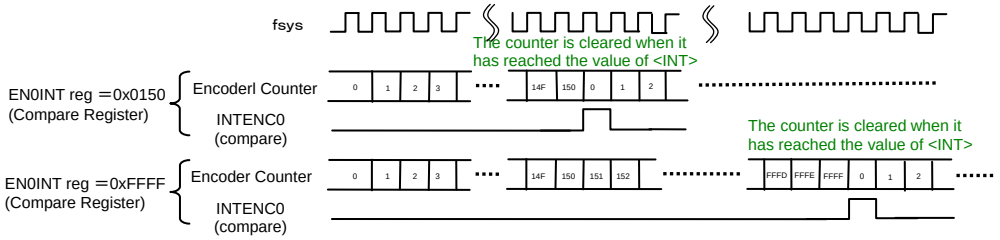
12.3.2.2 Sensor Event Count Mode



12.3.2.3 Sensor Timer Count Mode

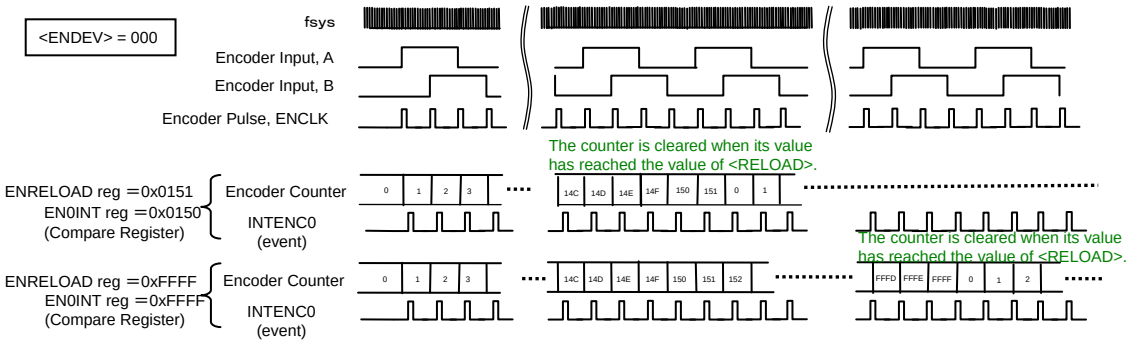


12.3.2.4 Timer Mode

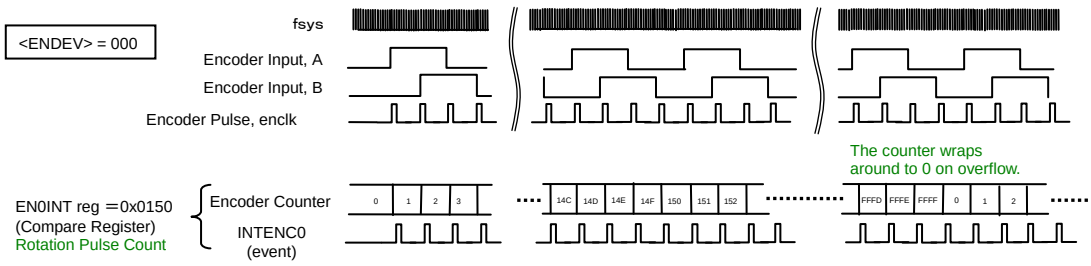


12.3.3 Counter Operation and Interrupt Generation When <CMPEN> = 0

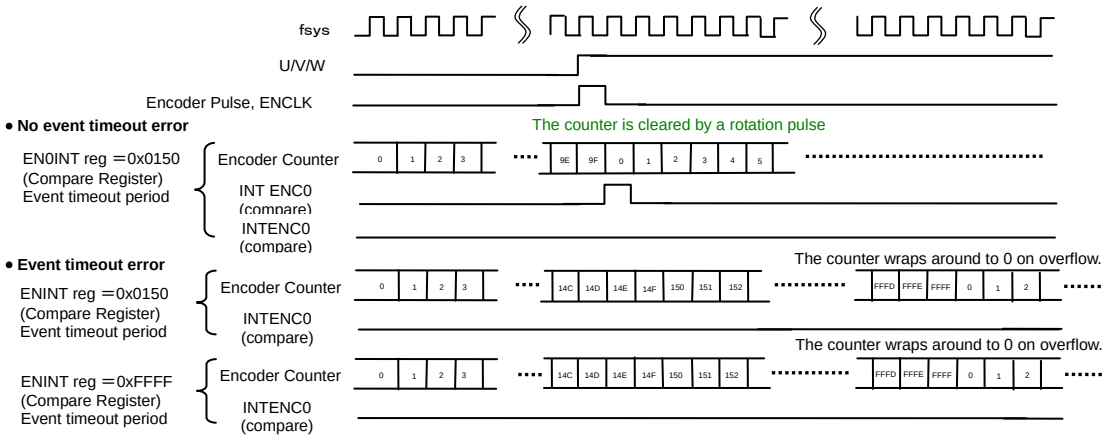
12.3.3.1 Encoder Mode



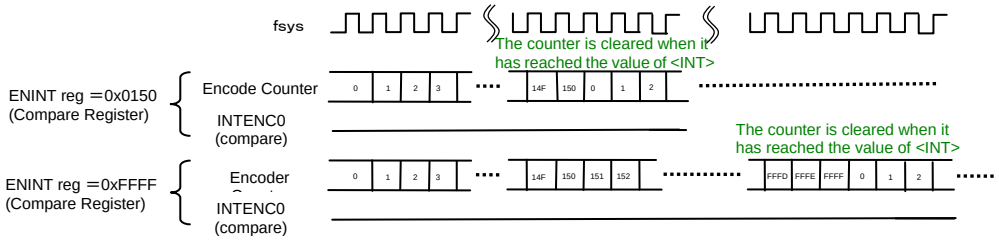
12.3.3.2 Sensor Event Count Mode



12.3.3.3 Sensor Timer Count Mode



12.3.3.4 Timer Mode



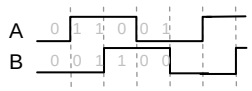
12.3.4 Encoder Rotation Direction

The following diagrams illustrate the phase shifting of the A, B and Z channels.

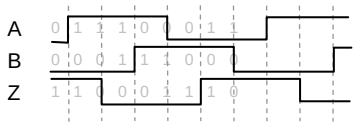
The ENC can interface with both two-phase (A/B) and three-phase (A/B/Z) encoder inputs. For three-phase encoder inputs, <P3EN> should be set to 1.

(1) Here are possible combinations of values of the A (U), B (V) and Z (W) signals during CW rotation.

・For two-phase inputs

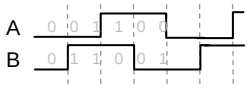


・For three-phase inputs

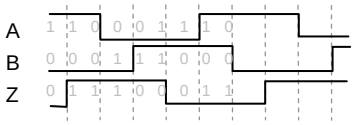


(2) Here are possible combinations of values of the A (U), B (V) and Z (W) signals during CCW rotation.

・For two-phase inputs



・For three-phase inputs



12.3.5 Counter Block

The counter block consists of a 24-bit up/down counter and its control logic.

12.3.5.1 Overview

The counter is configured as an up-counter or a down-counter, cleared and reloaded with a programmed value, according to the selected operating mode.

Table 12-4 summarizes how the counter is controlled.

Table 12-4 Counter Control

Operating Mode <MODE1:0>	<ZEN>	<P3EN>	Input Pins	Count	Up/Down	Counter Clear Conditions	Counter Reload Conditions	Counter Range (Reload Value)
Encoder Mode 00	0	0	A, B	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=<RELOAD>	-	0x0000 thru. <RELOAD>
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
	1		A, B, Z	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=<RELOAD> [3] Z trigger	-	
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
Sensor Event Count Mode 01	0	0	A, B	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=16'hFFFF	-	0x0000 thru. 0xFFFF
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
		1	A, B, Z	Encoder pulse (ENCLK)	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFF	-	
					Down	[1] <ENCLR> is set to 1.	[1] counter=0x0000	
Sensor Timer Count Mode 10	0	0	A, B	fsys	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFFFF [3] Encoder pulse (ENCLK)		0x000000 thru. 0xFFFFFF
		1	A, B, Z	fsys	Up			
Timer Mode 11	0	X	-	fsys	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFFFF [3] counter=<EN0INT>		0x000000 thru. 0xFFFFFF
	1		Z	fsys	Up	[1] <ENCLR> is set to 1. [2] counter=0xFFFFFF [3] counter=<EN0INT> [4] Z trigger		

Note: Clearing <ENRUN> to 0 does not clear the counter.
Setting <ENRUN> to 1 again causes the counter to restart from the current count.
The counter should be cleared by software setting <ENCLR> to 1.

12.3.6 Interrupts

The ENC has these interrupts: event (divided-clock/capture) interrupt, event timeout interrupt, timer compare interrupt and capture interrupt.

12.3.6.1 Overview

When <INTEN> = 1, the ENC generates interrupt requests, based on the counter value and the detection of a encoder pulse.

There are six interrupt sources, depending on the operating mode, and the settings of <CMPEN> and <ZEN>, as shown in Table 12-5.

Table 12-5 Interrupt Sources

	Interrupt Source	Description	Operating Mode	Interrupt Generation	Status Flag
1	Event counter interrupt	When <CMPEN> = 1, the encoderl counter counts events (encoder pulses). When it has reached the value programmed in <EN0INT>, an interrupt occurs.	Encoder mode and Sensor Event Count modes	When <INTEN> = 1 *and* <CMPEN> = 1	<CMP>
2	Event interrupt (Divided clock pulse)	An interrupt occurs on each divided clock pulse, which is derived by dividing the encoder pulse by a factor programmed in <ENDEV>.		When <INTEN> = 1	None
3	Event interrupt (Capture interrupt)	An interrupt occurs to indicate that an event (encoder pulse) has occurred, causing the counter value to be captured.	Sensor Timer Count mode	When <INTEN> = 1	None
4	Event timeout interrupt	When <CMPEN> = 1, the ENC uses a counter that counts up with fsys and is cleared by an event (encoder pulse). If no event occurs for a period of time programmed in <EN0INT>, an interrupt occurs.		When <INTEN> = 1 *and* <CMPEN> = 1	<CMP>
5	Timer compare interrupt	When <CMPEN> = 1, an interrupt occurs when the timer has reached the value programmed in <EN0INT>.	Timer mode	When <INTEN> = 1 *and* <CMPEN> = 1	<CMP>
6	Capture interrupt	An interrupt occurs when the counter value has been captured on an external trigger (Z input).		When <INTEN> = 1	None

In Sensor Timer Count mode and Timer mode, the value of the encoder counter can be captured into the ENCNT register.

The captured counter value can be read out of the ENCNT register.

In Sensor Timer Count mode, the value of the encoder counter is captured into the ENCNT register upon occurrence of an event (encoder pulse). The counter value can also be captured by writing a 1 to <SFTCAP>.

In Timer mode, the counter value can be captured by writing a 1 to <SFTCAP>. If <ZEN> is set to 1, the counter value can also be captured by an edge of the Z signal input selected via <ZESEL>.

13 Serial channel (UART/SIO)

Important
TMPM382 (64-pin version) does not implement SIO2 and SIO3.
Please do not use these functions if you use this product.

13.1 Features

This device has five serial I/O channels: SIO0 to SIO4.

Each channel operates in either the UART mode (asynchronous communication) or the I/O interface mode (synchronous communication) which is selected by the user.

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

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

In the above modes 1 and 2, parity bits can be added. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). Fig 13-2 shows the block diagram of SIO0.

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

As the SIOs 0 to 4 operate in the same way, only SIO0 is described here.

Table 13-1 Difference in the Specifications of SIO Modules

		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Pin name		TXD0 (PE0) RXD0 (PE1) $\overline{CTS0}$ /SCLK0 (PE2)	TXD1 (PA5) RXD1 (PA6) $\overline{CTS1}$ /SCLK1 (PA4)	TXD2 (PD5) RXD2 (PD6) $\overline{CTS2}$ /SCLK2 (PD4)	TXD3 (PF3) RXD3 (PF4) $\overline{CTS3}$ /SCLK3 (PF2)	TXD4 (PC6) RXD4(PC7) $\overline{CTS4}$ /SCLK4 (PC5)
Interrupt		INTRX0 INTTX0	INTRX1 INTTX1	INTRX2 INTTX2	INTRX3 INTTX3	INTRX4 INTTX4
In the UART mode, TMRB output to use for the serial transfer clock		TB4OUT (TMRB4)	TB4OUT (TMRB4)	TB7OUT (TMRB7)	TB7OUT (TMRB7)	MTTB0OUT (MPT0)
Register name (address)	Enable register	SC0EN 0x4002_0080	SC1EN 0x4002_00C0	SC2EN 0x4002_0100	SC3EN 0x4002_0140	SC4EN 0x4002_0180
	Transmit/ receive register	SC0BUF 0x4002_0084	SC1BUF 0x4002_00C4	SC2BUF 0x4002_0104	SC3BUF 0x4002_0144	SC4BUF 0x4002_0184
	Control register	SC0CR 0x4002_0088	SC1CR 0x4002_00C8	SC2CR 0x4002_0108	SC3CR 0x4002_0148	SC4CR 0x4002_0188
	Mode control register 0	SC0MOD0 0x4002_008C	SC1MOD0 0x4002_00CC	SC2MOD0 0x4002_010C	SC3MOD0 0x4002_014C	SC4MOD0 0x4002_018C
	Baud rate generator control	SC0BRCR 0x4002_0090	SC1BRCR 0x4002_00D0	SC2BRCR 0x4002_0110	SC3BRCR 0x4002_0150	SC4BRCR 0x4002_0190

		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
	Baud rate generator control 2	SC0BRADD 0x4002_0094	SC1BRADD 0x4002_00D4	SC2BRADD 0x4002_0114	SC3BRADD 0x4002_0154	SC4BRADD 0x4002_0194
	Mode control register 1	SC0MOD1 0x4002_0098	SC1MOD1 0x4002_00D8	SC2MOD1 0x4002_0118	SC3MOD1 0x4002_0158	SC4MOD1 0x4002_0198
	Mode control register 2	SC0MOD2 0x4002_009C	SC1MOD2 0x4002_00DC	SC2MOD2 0x4002_011C	SC3MOD2 0x4002_015C	SC4MOD2 0x4002_019C
	Receive FIFO configuration register	SC0RFC 0x4002_00A0	SC1RFC 0x4002_00E0	SC2RFC 0x4002_0120	SC3RFC 0x4002_0160	SC4RFC 0x4002_01A0
	Transmit FIFO configuration register	SC0TFC 0x4002_00A4	SC1TFC 0x4002_00E4	SC2TFC 0x4002_0124	SC3TFC 0x4002_0164	SC4TFC 0x4002_01A4
	Receive FIFO status register	SC0RST 0x4002_00A8	SC1RST 0x4002_00E8	SC2RST 0x4002_0128	SC3RST 0x4002_0168	SC4RST 0x4002_01A8
	Transmit FIFO status register	SC0TST 0x4002_00AC	SC1TST 0x4002_00EC	SC2TST 0x4002_012C	SC3TST 0x4002_016C	SC4TST 0x4002_01AC
	FIFO configuration register	SC0FCNF 0x4002_00B0	SC1FCNF 0x4002_00F0	SC2FCNF 0x4002_0130	SC3FCNF 0x4002_0130	SC4FCNF 0x4002_01B0

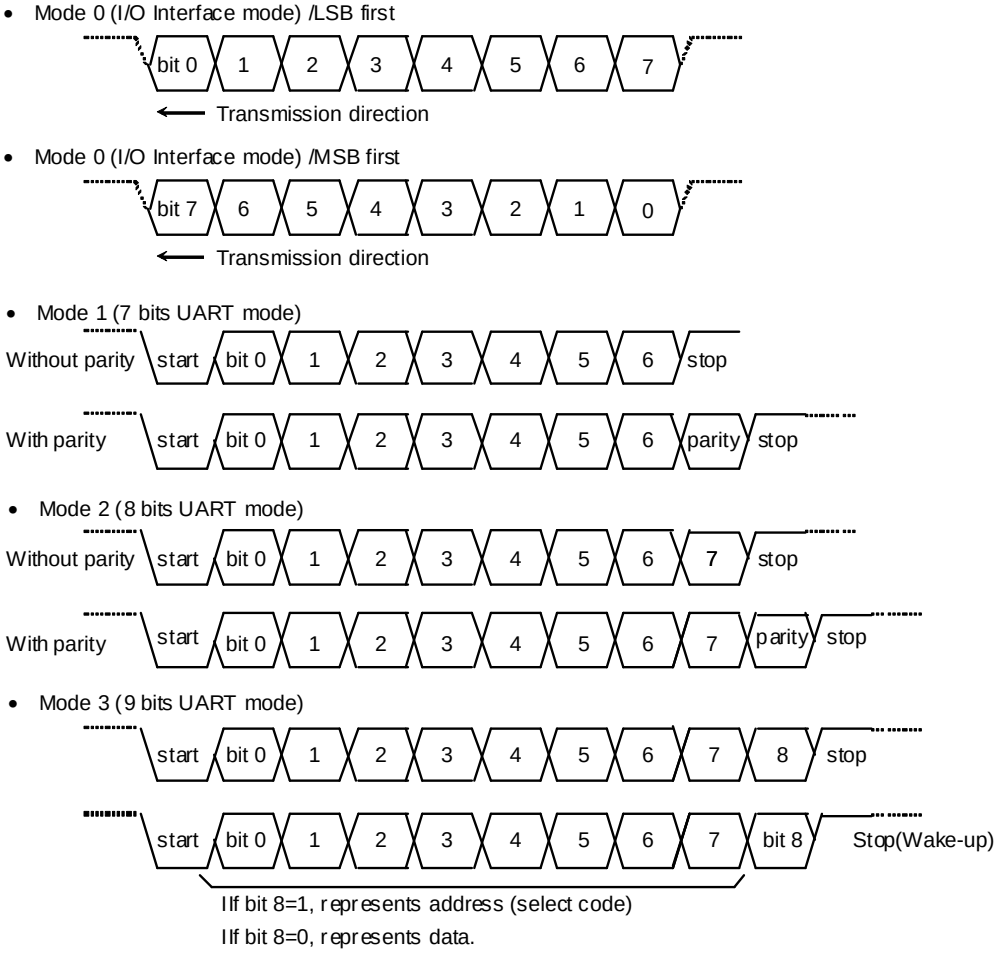


Fig 13-1 Data Format

13.2 Block Diagram (Channel 0)

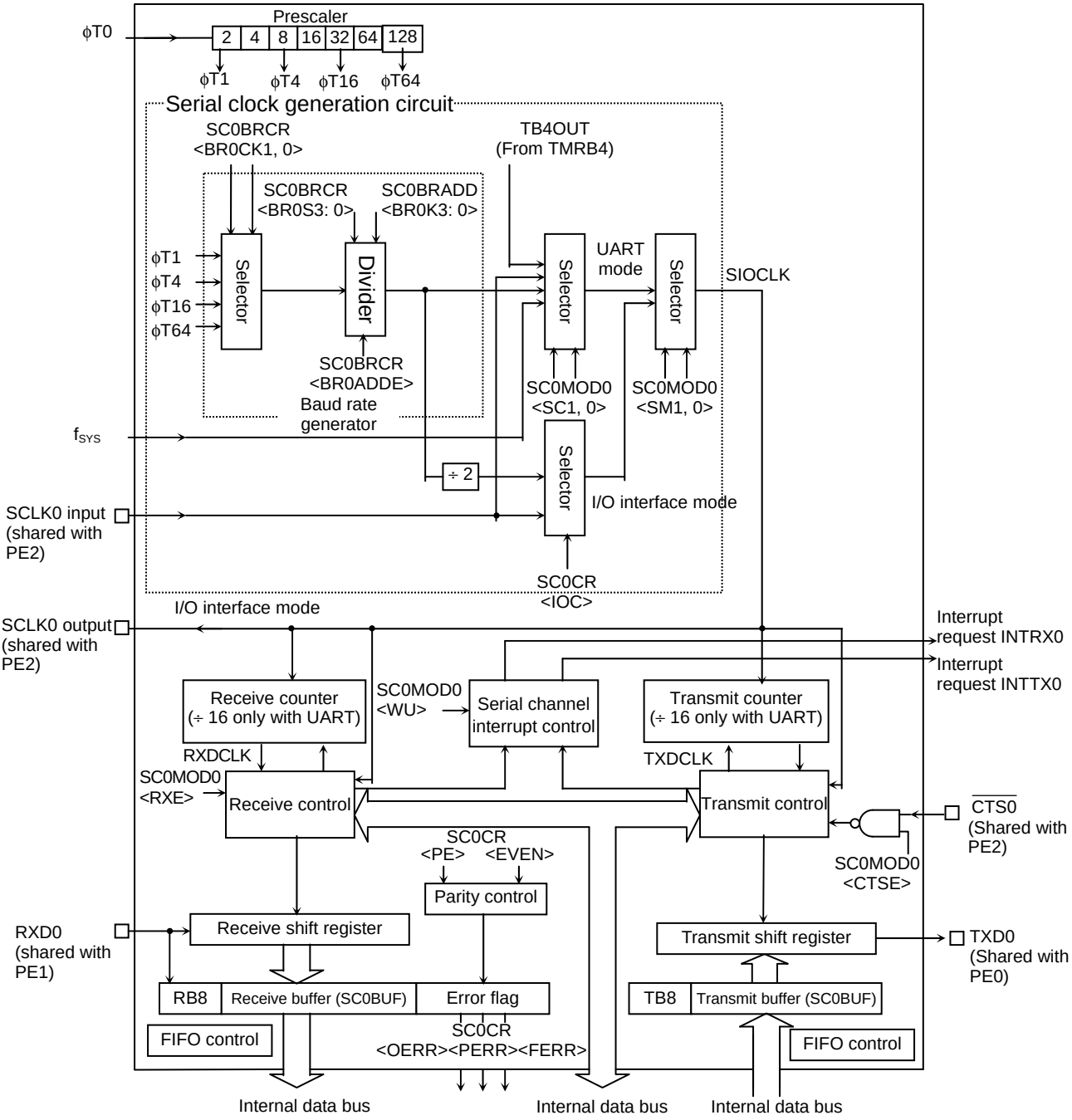


Fig 13-2 SIO0 Block Diagram

13.3 Operation of Each Circuit (Channel 0)

13.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock ϕT0 to the prescaler is selected by CGSYSCR1 of CG <PRCK2:0> to provide the frequency of either fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32.

The clock frequency fperiph is either the clock “fgear,” to be selected by CGSYSCR1<FPSEL> of CG, or the clock “fc” before it is divided by the clock gear.

The prescaler becomes active only when the baud rate generator is selected for generating the serial transfer clock. Table 13-1 list the prescaler output clock resolution.

Table 13-2 Clock Resolution to the Baud Rate Generator @ = 40MHz

Peripheral clock select <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000(fperiph/1)	fc/2^1(0.05μs)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		001(fperiph/2)	fc/2^2(0.1μs)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		010(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		011(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		100(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		101(fperiph/32)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
	100(fc/2)	000(fperiph/1)	fc/2^2(0.1μs)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		001(fperiph/2)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		010(fperiph/4)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		011(fperiph/8)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		100(fperiph/16)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
		101(fperiph/32)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)	fc/2^{13}(204.8μs)
	101(fc/4)	000(fperiph/1)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		001(fperiph/2)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		010(fperiph/4)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		011(fperiph/8)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
		100(fperiph/16)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)	fc/2^{13}(204.8μs)
		101(fperiph/32)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)	fc/2^{14}(409.6μs)
	110(fc/8)	000(fperiph/1)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		001(fperiph/2)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		010(fperiph/4)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
		011(fperiph/8)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)	fc/2^{13}(204.8μs)
		100(fperiph/16)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)	fc/2^{14}(409.6μs)
		101(fperiph/32)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)	fc/2^{13}(204.8μs)	fc/2^{15}(819.2μs)
1 (fc)	000 (fc)	000(fperiph/1)	fc/2^1(0.05μs)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		001(fperiph/2)	fc/2^2(0.1μs)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		010(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		011(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		100(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		101(fperiph/32)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
	100(fc/2)	000(fperiph/1)	−	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		001(fperiph/2)	fc/2^2(0.1μs)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		010(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		011(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		100(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		101(fperiph/32)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
	101(fc/4)	000(fperiph/1)	−	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		001(fperiph/2)	−	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		010(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		011(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		100(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		101(fperiph/32)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
	110(fc/8)	000(fperiph/1)	−	−	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		001(fperiph/2)	−	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		010(fperiph/4)	−	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		011(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		100(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		101(fperiph/32)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)

(Note 1) The prescaler output clock ϕTn must be selected so that the relationship "ϕTn < fsys: clock gear clock, one of fc,fc/2,fc/4,fc/8,fc/16" is satisfied (so that ϕTn is slower than fsys).
(Note 2) Do not change the clock gear while SIO is operating.
(Note 3) The horizontal lines in the above table indicate that the setting is prohibited.

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

13.3.2 Baud Rate Generator

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

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

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

The highest baud rate of each mode is limited.

- UART mode

  1) If SC0BRCR <BR0ADDE> = 0,

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

  2) If SC0BRCR <BR0ADDE> = 1,

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

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

- I/O interface mode

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

- Baud rate calculation to use the baud rate generator:

  1) UART mode

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

  The highest baud rate out of the baud rate generator is 1.25 Mbps.

  The fsys frequency, which is independent of the baud rate generator, can be used as the serial clock. In this case, the highest baud rate will be 2.5 Mbps.

  2) I/O interface mode

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

  When it uses a double buffer, the highest baud rate generated with the baud rate generator becomes 10Mbps. (If double buffering is not used, the highest baud rate will be 5.0 Mbps).

- Example baud rate setting:

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

Selecting fc = 39.321 MHz for fperiph, setting ϕT0 to fperiph/16, using the baud rate generator input clock ϕT1, setting the divide ratio N (SC0BRCR<BR0S3:0>) = 4, and setting SC0BRCR<BR0ADDE> = "0," the resulting baud rate in the UART mode is calculated as follows:

* Clocking conditions
  - System clock : High-speed (fc)
  - High speed clock gear : x 1 (fc)
  - Prescaler clock : f_{periph}/16 ($f_{periph} = f_{sys}$)

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

$$= 39.321 \times 10^6 \div 32 \div 4 \div 16 \quad 19200 \text{ (bps)}$$

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

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

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

* Clocking conditions
  - System clock : High-speed (fc)
  - High-speed clock gear : x 1 (fc)
  - Prescaler clock : f_{periph}/4 ($f_{periph} = f_{sys}$)

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

$$= 9.6 \times 10^6 \div 16 \div (7 + \frac{13}{16}) \div 16 = 4800 \text{ (bps)}$$

Also, an external clock input may be used as the serial clock. The resulting baud rate calculation is shown below:

- Baud rate calculation for an external clock input:

1) UART mode

Baud Rate = external clock input / 16

The period of the external clock input must be equal to or greater than 2/fsys.

and it is neccesary to set the highest baud rate less than 1.25 (Mbps).

2) I/O interface mode

Baud Rate = external clock input

When double buffering is used, it is necessary to satisfy the following relationship:

External clock input period > 6/fsys

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

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

External clock input period > 8/fsys

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

The baud rate examples for the UART mode are shown in Table 13-3 and Table 13-4 .

Table 13-3 Selection of UART Baud Rate

(Using the baud rate generator with SC0BRCR <BR0ADDE> = 0) Unit: (kbps)

fc [MHz]	Input clock / Divide ratio N (Set to SC0BRCR <BR0S3 : 0>)	ϕT1 (fc/4)	ϕT4 (fc/16)	ϕT16 (fc/64)	ϕT64 (fc/256)
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	0	9.600	2.400	0.600	0.150

(Note) This table shows the case where the system clock is set to fc, the clock gear is set to fc/1, and the prescaler clock is set to f_{periph}/2.

Table 13-4 Selection of UART Baud Rate

(The TMRB4 timer output (internal TB4OUT) is used with the timer input clock set to ϕT1.)

Unit: (Kbbs)

TB4RG0 \ fc	40 MHz	9.8304 MHz	8 MHz
0x0001	312.5	76.8	62.5
0x0002	156.25	38.4	31.25
0x0003		25.6	
0x0004	78.125	19.2	15.625
0x0005	62.5	15.36	12.5
0x0006		12.8	
0x0008	39.0625	9.6	
0x000A	31.25	7.68	6.25
0x0010	19.53125	4.8	
0x0014	15.625	3.84	3.125

Baud rate calculation to use the TMRB4 timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by CGSYSCR0<PRCK2:0>}}{(\text{TBxRG0}\times 2)\times 2\times 16}$$

(When input clock to the timer TMRB4 is ϕT1)

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

(Note 1) In the I/O interface mode, the TMRB4 timer output signal cannot be used internally as the transfer clock.

(Note 2) This table shows the case where the system clock is set to fc, the clock gear is set to fc, and the prescaler clock is set to f_{periph}/2.

13.3.3 Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

- I/O interface mode

  In the SCLK output mode with the SC0CR <IOC> serial control register set to “0,” the output of the previously mentioned baud rate generator is divided by 2 to generate the basic clock.

  In the SCLK input mode with SC0CR <IOC> set to “1,” rising and falling edges are detected according to the SC0CR <SCLKS> setting to generate the basic clock.

- Asynchronous (UART) mode :

  According to the settings of the serial control mode register SC0MOD0 <SC1:0>, either the clock from the baud rate register, the system clock (f_{SYS}), the internal output signal of the TMRB4 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

13.3.4 Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by SIOCLK. Sixteen SIOCLK clock pulses are used in receiving a single data bit while the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

13.3.5 Receive Control Unit

- I/O interface mode:

  In the SCLK output mode with SC0CR <IOC> set to “0,” the RXD0 pin is sampled on the rising edge of the shift clock output to the SCLK0 pin.

  In the SCLK input mode with SC0CR <IOC> set to “1,” the serial receive data RXD0 pin is sampled on the rising or falling edge of SCLK input depending on the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

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

13.3.6 Receive Buffer

The receive buffer is of a dual structure to prevent overrun errors. The receive shift register stores the received data bit-by-bit. When a complete set of bits have been stored, they are moved to the receive buffer (SC0BUF). At the same time, the receive buffer full flag (SC0MOD2 “RBFLL”) is set to “1” to indicate that valid data is stored in the receive buffer. However, if the receive FIFO is set enabled, the receive data is moved to the receive FIFO and this flag is immediately cleared.

If the receive FIFO has been disabled (SCOFCNF <CNFG> = 0 and SC0MOD1<FDPX1:0> =01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1<FDPX1:0> = 01), an interrupt will be generated according to the SC0RFC <RIL1:0> setting.

The CPU will read the data from either the receive buffer (SC0BUF) or from the receive FIFO (the address is the same as that of the receive buffer). If the receive FIFO has not been enabled, the receive buffer full flag <RBFLL> is cleared to “0” by the read operation. The next data received can be stored in the receive shift register even if the CPU has not read the previous data from the receive buffer (SC0BUF) or the receive FIFO.

If SCLK is set to generate clock output in the I/O interface mode, the double buffer control bit SC0MOD2 <WBUF> can be programmed to enable or disable the operation of the receive buffer (SCOBUF).

By disabling the receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (SCOFCNF <CNFG> = 0 and <FDPX1:0> = 01), handshaking with the other side of communication can be enabled and the SCLK output stops each time one frame of data is transferred. In this setting, the CPU reads data from the receive shift register. By the read operation of CPU, the SCLK output resumes.

If the receive buffer (i.e., double buffering) is enabled but the receive FIFO is not enabled, the SCLK output is stopped when the first receive data is moved from the receive shift register to the receive buffer and the next data is stored in the first buffer filling both buffers with valid data. When the receive buffer is read, the data of the receive shift register is moved to the receive buffer and the SCLK output is resumed upon generation of the receive interrupt INTRX0. Therefore, no buffer overrun error will be caused in the I/O interface SCLK output mode regardless of the setting of the double buffer control bit SC0MOD2 <WBUF>.

If the receive buffer (double buffering) is enabled and the receive FIFO is also enabled (SCNFCNF <CNFG> = 1 and <FDPX1:0> = 01/11), the SCLK output will be stopped when the receive FIFO is full (according to the setting of SCOFNCF <RFST>) and receive buffer and receive shift register contain valid data. Also in this case, if SCOFCNF <RXTXCNT> has been set to “1,” the receive control bit RXE will be automatically cleared upon suspension of the SCLK output. If it is set to “0,” automatic clearing will not be performed.

(Note) In this mode, the SC0CR <OEER> flag is insignificant and the operation is undefined. Therefore, before switching from the SCLK output mode to another mode, the SC0CR register must be read to initialize this flag.

In other operating modes, the operation of the receive buffer is always valid, thus improving the performance of continuous data transfer. If the receive FIFO is not enabled, an overrun error occurs when the data in the receive buffer (SC0BUF) has not been read before the receive shift register is full with the next receive data. If an overrun error occurs, data in the receive shift register will be lost while data in the receive buffer and the contents of SC0CR <RB8> remain intact. If the receive FIFO is enabled, the FIFO must be read before the FIFO is full and the receive buffer is written by the next data through receive shift register. Otherwise, an overrun error will be generated and the receive FIFO overrun error flag will be set. Even in this case, the data already in the receive FIFO remains intact.

The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SC0CR <RB8>.

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

13.3.7 Receive FIFO Buffer

In addition to the double buffer function already described, data may be stored using the receive FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte receive buffer can be enabled. Also, in the UART mode or I/O interface mode, data may be stored up to a predefined fill level. When the receive FIFO buffer is to be used, be sure to enable the double buffer function.

If data with parity bit is to be received in the UART mode, parity check must be performed each time a data frame is received.

13.3.8 Receive FIFO Operation

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

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

① I/O interface mode with SCLK output:

The following example describes the case a 4-byte data stream is received in the half duplex mode:

SCxMOD1<6:5>=01 : Transfer mode is set to half duplex mode

SC0FCNF <4:0>=10111: Automatically inhibits continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

SC0RFC<1:0>=00: The fill level of FIFO in which generated receive interrupt is set to 4-byte..

SC0RFC<7:6>=11: Clears receive FIFO and sets the condition of interrupt generation.

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

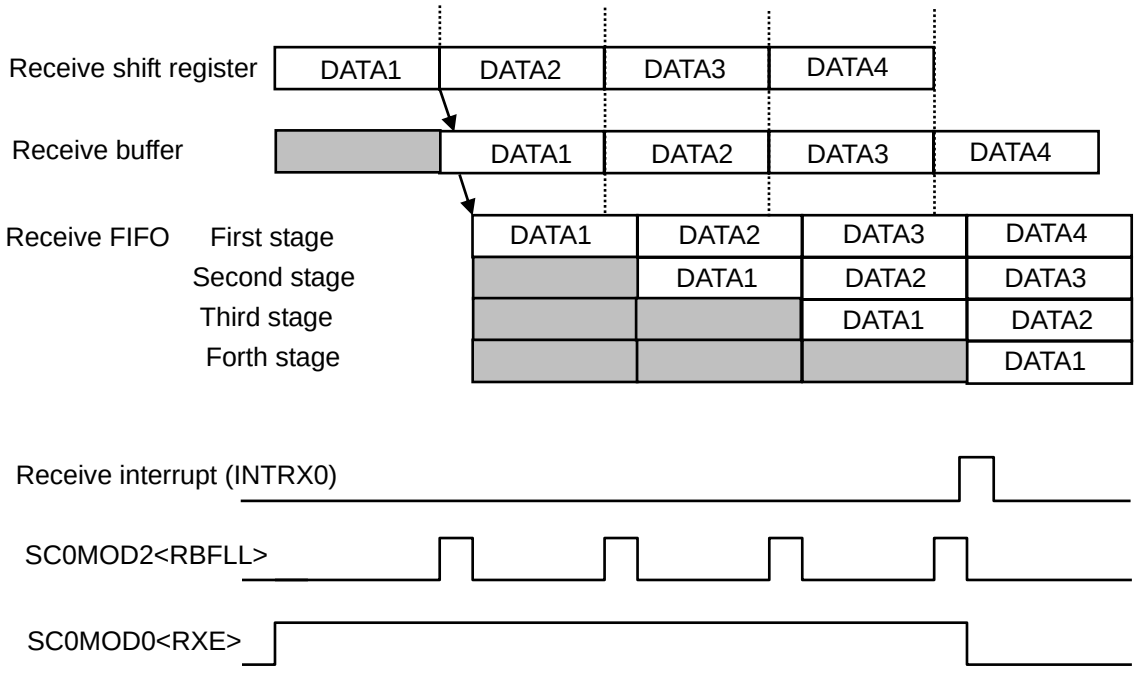


Fig 13-3 Receive FIFO Operation (1)

② I/O interface mode with SCLK input:

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

SC0MOD1 <6:5> = 01: Transfer mode is set to half duplex mode

SC0FCNF <4:0> = 10101: Automatically allows continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the maximum.

SC0RFC <1:0> = 00: The fill level of FIFO in which generated receive interrupt is set to 4-byte..

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

The number of bytes to be used in the receive FIFO is the maximum allowable number.

In this condition, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing “1” to the RXE bit. After receiving 4 bytes, receive FIFO interrupt is generated. This setting enables the next data reception as well. The next 4 bytes can be received before all the data is read from FIFO.

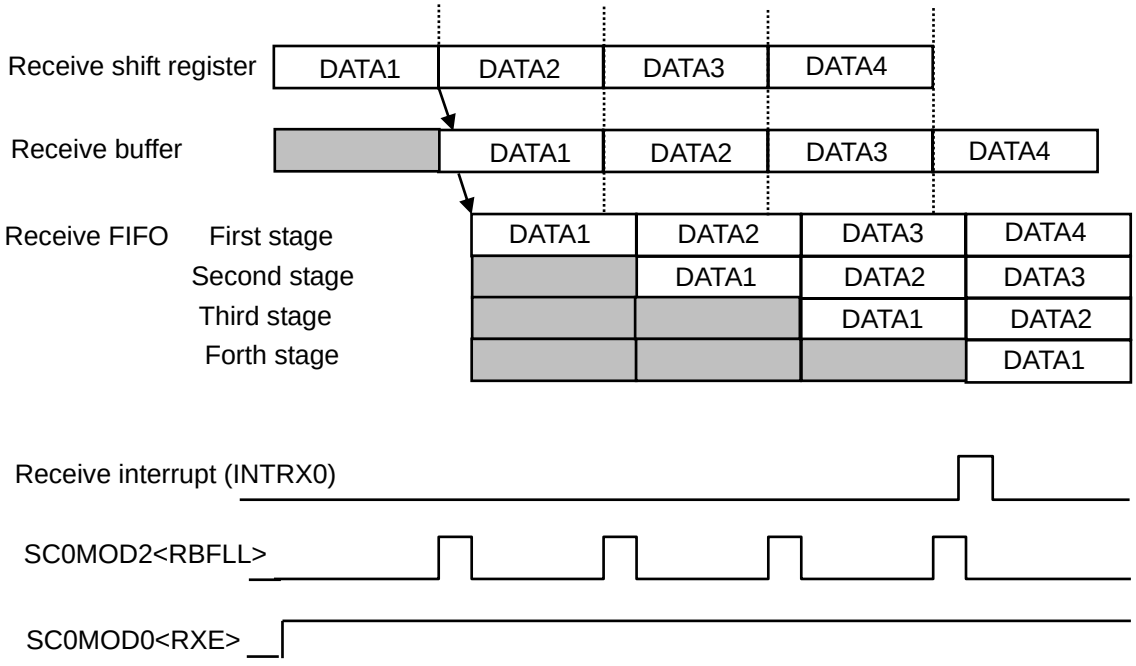


Fig 13-4 Receive FIFO Operation (2)

13.3.9 Transmit Counter

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

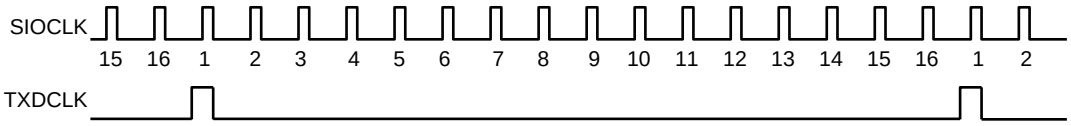


Fig 13-5 Transmit Clock Generation

13.3.10 Transmit Control Unit

- I/O interface mode:

  In the SCLK output mode with SC0CR <IOC> set to “0,” each bit of data in the transmit buffer is output to the TXD0 pin on the falling edge of the shift clock output from the SCLK0 pin.

  In the SCLK input mode with SC0CR <IOC> set to “1,” each bit of data in the transmit buffer is output to the TXD0 pin on the rising or falling edge of the input SCLK signal according to the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

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

- Handshake function

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

When the $\overline{CTS0}$ pin is set to the “H” level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{CTS0}$ pin returns to the “L” level. However in this case, the INTTX0 interrupt is generated, the next transmit data is requested to the CPU, data is written to the transmit buffer, and it waits until it is ready to transmit data.

Although no $\overline{RTS}$ pin is provided, a handshake control function can be easily implemented by assigning a port for the $\overline{RTS}$ function. By setting the port to “H” level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

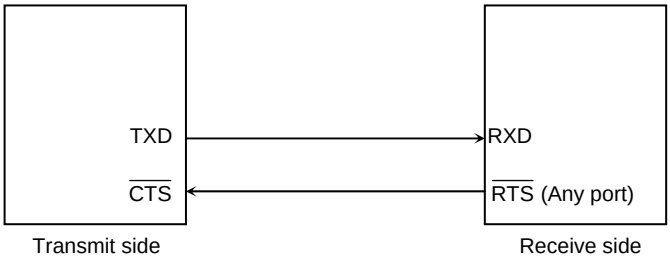


Fig 13-6 Handshake Function

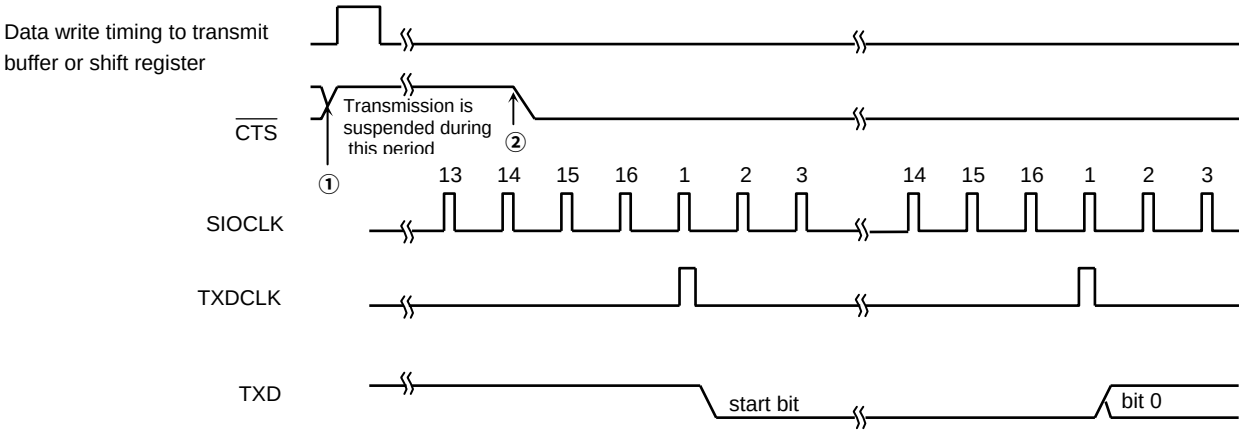


Fig 13-7 $\overline{CTS}$ (Clear to Transmit) Signal Timing

(Note 1)	**If the $\overline{CTS}$ signal is set to “H” during transmission, the next data transmission is suspended after the current transmission is completed.**
(Note 2)	**Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{CTS}$ is set to “L.”**

13.3.11 Transmit Buffer

The transmit buffer (SC0BUF) is in a dual structure. The double buffering function may be enabled or disabled by setting the double buffer control bit <WBUF> in serial mode control register 2 (SC0MOD2). If double buffering is enabled, data written to Transmit Buffer (SCOBUF) is moved to Transmit shift register.

If the transmit FIFO has been disabled (SCOFCNF <CNFG> = 0 or 1 and SC0MOD1＜FDPX1:0＞＝01), the INTTX0 interrupt is generated at the same time and the transmit buffer empty flag <TBEMP> of SC0MOD2 is set to “1.” This flag indicates that Transmit Buffer is now empty and that the next transmit data can be written. When the next data is written to Transmit Buffer, the <TBEMP> flag is cleared to “0.”

If the transmit FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1＜FDPX1:0＞＝10/11), any data in the transmit FIFO is moved to the Transmit Buffer and <TBEMP> flag is immediately cleared to “0.” The CPU writes data to Transmit Buffer or to the transmit FIFO.

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in Transmit Buffer before the next frame clock input, which occurs upon completion of data transmission from Transmit shift register, an under-run error occurs and a serial control register (SC0CR) <PERR> parity/under-run flag is set.

If the transmit FIFO is enabled in the I/O interface SCLK input mode, when data transmission from Transmit shift register is completed, the Transmit Buffer data is moved to Transmit shift register and any data in transmit FIFO is moved to Transmit Buffer at the same time.

If the transmit FIFO is disabled in the I/O interface SCLK output mode, when data in Transmit Buffer is moved to Transmit shift register and the data transmission is completed, the SCLK output stops. So, no under-run errors can be generated.

If the transmit FIFO is enabled in the I/O interface SCLK output mode, the SCLK output stops upon completion of data transmission from Transmit shift register if there is no valid data in the transmit FIFO.

Note) In the I/O interface SCLK output mode, the SC0CR <PEER> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the SCLK output mode to another mode, SC0CR must be read in advance to initialize the flag.

If double buffering is disabled, the CPU writes data only to Transmit shift register and the transmit interrupt INTTX0 is generated upon completion of data transmission.

If handshaking with the other side is necessary, set the double buffer control bit <WBUF> to “0” (disable) to disable Transmit Buffer; any setting for the transmit FIFO should not be performed.

13.3.12 Transmit FIFO Buffer

In addition to the double buffer function already described, data may be stored using the transmit FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte transmit buffer can be enabled. In the UART mode or I/O interface mode, up to 4 bytes of data may be stored.
If data is to be transmitted with a parity bit in the UART mode, parity check must be performed on the receive side each time a data frame is received.

13.3.13 Transmit FIFO Operation

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

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

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

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

SCxMOD1<6:5> 10 : Transfer mode is set to half duplex.

SCxFCNF<4:0> 01011 : Transmission is automatically disabled if FIFO becomes empty.
The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

SCxTFC<1:0> 00 : Sets the interrupt generation fill level to "0".

SCxTFC<7:6> 11 : Clears receive FIFO and sets the condition of interrupt generation.

In this condition, data transmission can be initiated by setting the transfer mode to half duplex, writing 5 bytes of data to the transmit FIFO, and setting the SC0MOD1<TXE> bit to “1.” When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.

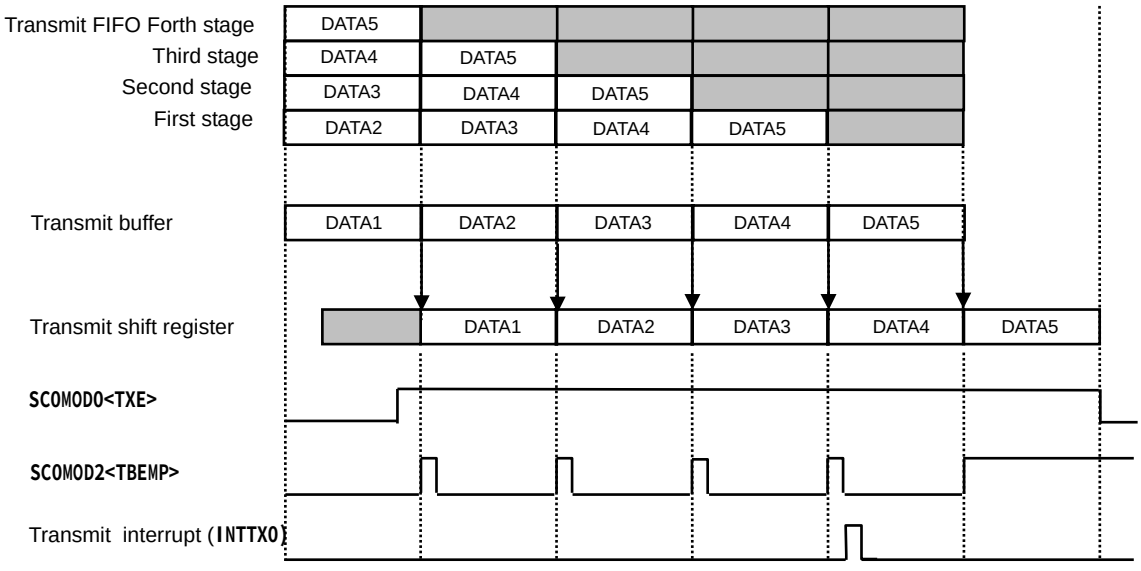


Fig 13-8 Transmit FIFO Operation(1)

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

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

SCxMOD1<6:5> 10 : Transfer mode is set to half duplex.

SCxFCNF<4:0> 01011 : Transmission is automatically disabled if FIFO becomes empty.
The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

SCxTFC<1:0> 00 : Sets the interrupt generation fill level to "0".

SCxTFC<7:6> 11 : Clears receive FIFO and sets the condition of interrupt generation.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 5 bytes of data to the transmit FIFO, and setting the <TXE> bit to “1.” When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated.

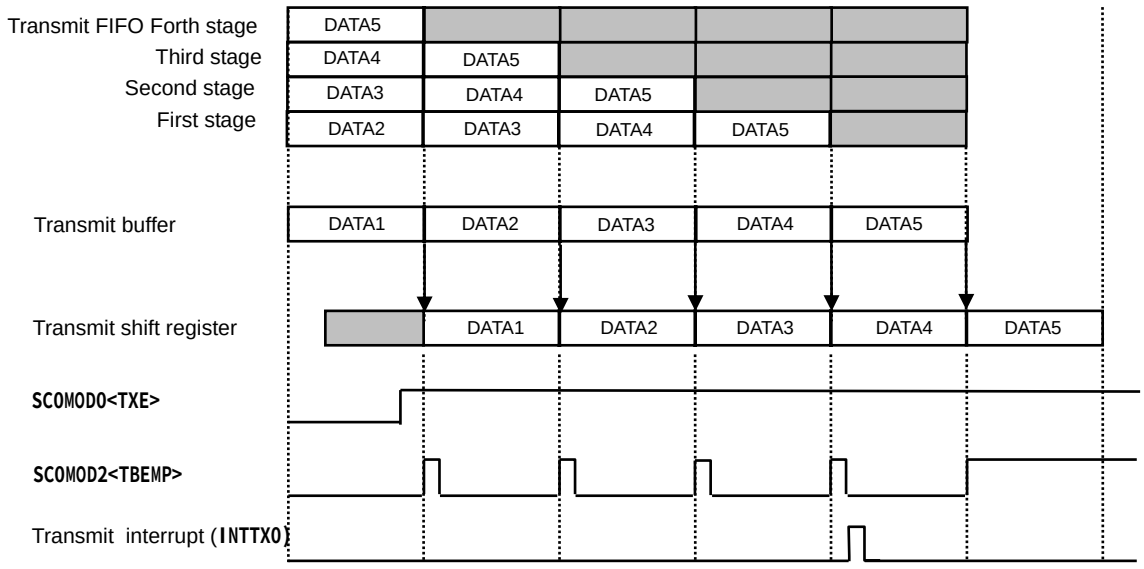


Fig 13-9 Transmit FIFO Operation(2)

13.3.14 Parity Control Circuit

If the parity addition bit <PE> of the serial control register SC0CR is set to “1,” data is sent with the parity bit. Note that the parity bit may be used only in the 7- or 8-bit UART mode. The <EVEN> bit of SC0CR selects either even or odd parity.

Upon data transmission, the parity control circuit automatically generates the parity with the data written to the transmit buffer (SC0BUF). After data transmission is complete, the parity bit will be stored in SC0BUF bit 7 <TB7> in the 7-bit UART mode and in bit 7 <TB8> in the serial mode control register SC0MOD in the 8-bit UART mode. The <PE> and <EVEN> settings must be completed before data is written to the transmit buffer.

Upon data reception, the parity bit for the received data is automatically generated while the data is shifted to receive shift register and moved to receive buffer (SC0BUF). In the 7-bit UART mode, the parity generated is compared with the parity stored in SC0BUF <RB7>, while in the 8-bit UART mode, it is compared with the bit 7 <RB8> of the SC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the SC0CR register is set.

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

In the I/O interface mode, the SC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

13.3.15 Error Flag

Three error flags are provided to inprove the reliability of received data.

1. Overrun error <OERR>: Bit 4 of the serial control register SC0CR

   In both UART and I/O interface modes, this bit is set to “1” when an error is generated by completing the reception of the next frame receive data before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied). This flag is set to “0” when it is read. In the I/O interface SCLK output mode, no overrun error is generated and therefore, this flag is inoperative and the operation is undefined.

Note: To switch the I/O interface SCLK output mode to other modes, read the SCxCR register and clear the overrun flag.

2. Parity error/under-run error <PERR>: Bit 3 of the SC0CR register

In the UART mode, this bit is set to “1” when a parity error is generated. A parity error is generated when the parity generated from the received data is different from the parity received. This flag is set to “0” when it is read.

In the I/O interface mode, this bit indicates an under-run error. When the double buffer control bit <WBUF> of the serial mode control register SC0MOD2 is set to “1” in the SCLK input mode, if no data is set to the transmit double buffer before the next data transfer clock after completing the transmission from the transmit shift register, this error flag is set to “1” indicating an under-run error. If the transmit FIFO is enabled, any data content in the transmit FIFO will be moved to the buffer. When the transmit FIFO and the double buffer are both empty, an under-run error will be generated. Because no under-run errors can be generated in the SCLK output mode, this flag is inoperative and the operation is undefined. If Transmit Buffer is disabled, the under-run flag <PERR> will not be set. This flag is set to “0” when it is read.

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

3. Framing error <FERR>: Bit 2 of the SC0CR register

In the UART mode, this bit is set to “1” when a framing error is generated. This flag is set to “0” when it is read. A framing error is generated if the corresponding stop bit is determined to be “0” by sampling the bit at around the center. Regardless of the <SBLEN> (stop bit length) setting of the serial mode control register 2, SC0MOD2, the stop bit status is determined by only 1 bit on the receive side.

Operation mode	Error flag	Function
UART	OERR	Overrun error flag
	PERR	Parity error flag
	FERR	Framing error flag
I/O Interface (SCLK input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O Interface (SCLK output)	OERR	Operation undefined
	PERR	Operation undefined
	FERR	Fixed to 0

13.3.16 Direction of Data Transfer

In the I/O interface mode, the direction of data transfer can be switched between “MSB first” and “LSB first” by the data transfer direction setting bit <DRCHG> of the SC0MOD2 serial mode control register 2. Don't switch the direction when data is being transferred.

13.3.17 Stop Bit Length

In the UART transmission mode, the stop bit length can be set to either 1 or 2 bits by bit 4 <SBLEN> of the SC0MOD2 register. The length of the STOP bit data is determined as one-bit when it is received regardless of the setting of this bit.

13.3.18 Status Flag

If the double buffer function is enabled (SC0MOD2 <WBUF> = “1”), the bit 6 flag <RBFLL> of the SC0MOD2 register indicates the condition of receive buffer full. When one frame of data has been received and transferred from shift registers to buffers, this bit is set to “1” to show that buffers are full (data is stored in the buffers). When the receive buffer is read by CPU/DMAC, it is cleared to “0.” If <WBUF> is set to “0,” this bit is insignificant and must not be used as a status flag. When double buffering is enabled (SC0MOD2 <WBUF> = “1”), the bit 7 flag <TBEMP> of the SC0MOD2 register indicates that Transmit Buffer is empty. When data is moved from Transmit Buffer to Transmit shift register, this bit is set to “1” indicating that Transmit Buffer is now empty. When data is set to the transmit buffer by CPU/DMAC, the bit is cleared to “0.” If <WBUF> is set to “0,” this bit is insignificant and must not be used as a status flag.

13.3.19 Configurations of Transmit/Receive Buffer

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

13.3.20 Software reset

Software reset is generated by writing the bits 1 and 0 of SC0MOD2 <SWRST1:0> as “10” followed by “01”. As a result, SC0MOD0<RXE>, SC0MOD1<TXE>, SC0MOD2<TBEMP>,<RBFLL>,<TXRUN> of mode registers and SC0CR<OERR>, <PERR>, <FERR> of control registers and internal circuit is initialized. Other states are maintained.

13.3.21 Signal Generation Timing

① UART Mode:

Receive Side

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

Transmit Side

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

I/O interface mode:

Receive Side

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

Transmit Side

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

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

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

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

13.4 Register Description (Only for Channel 0)

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

13.4.1 Enable register

SC0EN

	7	6	5	4	3	2	1	0
bit Symbol								SIOE
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							SIO operation 0:disabled 1:enabled

<SIOE>: Specified the SIO operation.
To use the SIO, enable the SIO operation.
When the operation is disabled, no clock is supplied to the other registers in the SIO module. This can reduce the power consumption.
If the SIO operation is executed and then disabled, the settings will be maintained in each register.

13.4.2 Buffer register

SC0BUF works as a transmit buffer for WR operation and as a receive buffer for RD operation.

SC0BUF

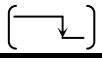
	7	6	5	4	3	2	1	0
bit Symbol	TB7/RB7	TB6/RB6	TB5/RB5	TB4/RB4	TB3/RB3	TB2/RB2	TB1/RB1	TB0/RB0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	TB7~0 : Transmit buffer/FIFO RB7~0 : Receive buffer/FIFO							

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

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

13.4.3 Control register

SC0CR

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to “0” when read)			R/W	
After reset	0	0	0	0	0	0	0	0
Function	Receive data bit 8 **(For UART)**	Parity **(For UART)** 0: Odd 1: Even	Add parity **(For UART)** 0: Disabled 1: Enabled	0: Normal operation 1: Error			0: SCLK0  1: SCLK0 	**(For I/O interface)** 0:Baud rate generator 1:SCLK0 pin input
				Overrun	Parity/ underrun	Framing		

<RB8>: 9th bit of the received data in the 9 bits UART mode.

<EVEN>: Selects even or odd parity.
“0”: odd parity.
“1”: even parity.
The parity bit may be used only in the 7- or 8-bit UART mode.

<PE>: Controls enabling/ disabling parity.
The parity bit may be used only in the 7- or 8-bit UART mode.

<OERR>: Error flag **(see note)**
<PERR>: Indicate overrun error, parity error, underrun error and framing error.
<FERR>:

<SCLKS>: Selects edge for data transmission and reception.
“0”: Data transmit/receive at rising edges of SCLK0
“1”: Data transmit/receive at falling edges of SCLK0

<IOC>: Selects input clock in the I/O interface mode.
“0”: baud rate generator
“1”: SCLK0 pin input.

(Note) Any error flag is cleared when read.

13.4.4 Mode control register 0

SC0MOD0

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transmit data bit 8	Handshake function control 0: CTS diable 1: CTS enable	Receive control 0:Reception disabled 1:Reception enabled	Wake-up function 0:Reception disabled 1:Reception enabled	Serial transfer mode 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode		Serial transfer clock **(for UART)** 00: TMRB,MPT output 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock (SCLK0 input)	

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

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

<RXE>: Controls reception **(see note).**
Set <RXE> after setting each mode register (SC0MOD0, SC0MOD1 and SC0MOD2).

<WU>: Controls wake-up function.
This function is available only at 9-bit UART mode.

	9-bit UART mode	Other modes
0	Interrupt when received	don't care
1	Interrupt only when RB9=1	

<SM1:0>: Specifies transfer mode.

<SC1:0>: Selects the serial transfer clock in the UART mode.
As for the I/O interface mode, the serial transfer clock can be set in the control register SC0CR.
When TMRB,MPT output is used for the serial transfer clock, TB4OUT is used in SIO0,1, TB7OUT is used in SIO2,3, and MTTB0OUT is used in SIO4.

(Note 1) With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2). Then set <RXE> to "1."
(Note 2) Do not stop the receive operation (by setting SCxMOD0<RXE> = "0") when data is being received.

13.4.5 Mode control register 1

SC0MOD1

	7	6	5	4	3	2	1	0
bit Symbol	I2SC	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	IDLE 0: Stop 1: Start	Transfer mode setting 00: Transfer prohibited 01: Half duplex(RX) 10:Half duplex(TX) 11:Full duplex		Transmit control 0:Disabled 1: Enabled	Interval time of continuous transmission **(for I/O interface)** 000: None 100:8SCLK 001:1SCLK 101:16SCLK 010:2SCLK 110:32SCLK 011:4SCLK 111:64SCLK			Write “0”.

<I2SC>: Specifies the IDLE mode operation.

<FDPX1:0>: Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<TXE>: This bit enables transmission and is valid for all the transfer modes (**see note)**. If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<SINT2:0>: Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode. This parameter is valid only for the I/O interface mode when SCLK0 pin input is not selected.

(Note 1) Specify the mode first and then specify the <TXE> bit.
(Note 2) Do not stop the transmit operation (by setting <TXE> = "0") when data is being transmitted.

13.4.6 Mode control register 2

SC0MOD2

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R			R/W				
After reset	1	0	0	0	0	0	0	0
Function	Transmit buffer empty flag 0: full 1: Empty	Receive Buffer full flag 0: Empty 1: full	In transmission flag 0: Stop 1: Start	STOP bit **(for UART)** 0: 1-bit 1: 2-bit	Setting transfer direction 0: LSB first 1: MSB first	W-buffer 0: Disabled 1: Enabled	SOFT RESET Overwrite “01” on “10”to reset.	

<TBEMP>: This flag shows that the transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to “1.” Writing data again to the double buffers sets this bit to “0.”
If double buffering is disabled, this flag is insignificant.

<RBFLL>: This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to “1” while reading this bit changes it to “0.”
If double buffering is disabled, this flag is insignificant.

<TXRUN>: This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> bits indicate the following status.

<TXRUN>	<TBEMP>	Status
1	-	Transmission in progress
0	1	Transmission completed
	0	Wait state with data in TX buffer

<SBLEN>: This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, it is fixed to LSB first.

<WBUF>: This parameter enables or disables the transmit/receive buffers to transmit (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART. When receiving data in the I/O interface mode (I SCLK input) and UART mode, double buffering is enabled in both cases that 0 or 1 is set to <WBUF> bit.

<SWRST1:0>: Overwriting “01” in place of “10” generates a software reset. When this software reset is executed, the following bits, transmitter, reciever and FIFO are initialized **(see note 1, 2 and 3)**.

Register	Bit
SC0MOD0	RXE
SC0MOD1	TXE
SC0MOD2	TBEMP,RBFLL,TXRUN,
SC0CR	OERR,PERR,FERR

(Note 1)	**While data transmission is in progress, any software reset operation must be executed twice in succession.**
(Note 2)	**A software reset requires 2 clocks-duration at the time between the end of recognition and the start of execution of software reset instruction.**
(Note 3)	**A software reset initializes other bits. Resetting a mode register and a control register are needed.**

13.4.7 Baud rate generator control register(SC0BRCR)
Baud rate generator control register 2(SC0BRADD)

SC0BRCR

	7	6	5	4	3	2	1	0
bit Symbol	-	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Write "0".	N + (16 – K)/ 16 divider function 0: disabled 1: enabled	Select input clock to the baud rate generator 00: φT1 01: φT4 10: φT16 11: φT64		Division ratio "N" 0000: 16 0001: 1 0010: 2 : 1111: 15			

SC0BRADD

	7	6	5	4	3	2	1	0
bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	"0" is read.				Specify K for the "N + (16 – K)/16" division 0000: Prohibited 0001: K=1 0010: K=2 : 1111: K=15			

<RB0ADDE>: Specifies N + (16-K)/16 division function.
N + (16-K)/16 division function can only be used in the UART mode.

<RB0CK1:0>: Specifies the baud rate generator input clock.

<RB0S3:0>: Specifies division ratio "N".

<RB0K3:0>: Specifies K for the "N+(16-K)/16" division.

The division ratio of the baud rate generator can be specified in the registers shown above. Table 13-5 lists the settings of baud rate generator division ratio.

Table 13-5 Setting division ratio

	BR0ADDE=0	BR0ADDE=1 **(Note 1)** (Only UART)
BR0S	Specify "N" **(Note 2) (Note 3)**	
BR0K	No setting required	Setting "K" **(Note 4)**
Division ratio	Divide by N	$N + \frac{(16-K)}{16}$ division

(Note 1) **To use the “N + (16 - K)/16” division function, be sure to set BR0K <BR0ADDE> to “1” after setting the K value to BR0K. The “N + (16 - K)/16” division function can only be used in the UART mode.**

(Note 2) **As a division ratio, 1 ("0001") or 16 ("0000") can not be applied to N when using the "N + (16 - K)/16"division function in the UART mode.**

(Note 3) **The division ratio "1" of the baud rate generator can be specified only when the double buffering is used in the I/O interface mode.**

(Note 4) **Specifying “K = 0” is prohibited.**

13.4.8 FIFO configuration register

SC0FCNF

	7	6	5	4	3	2	1	0
bit Symbol	Reserved	Reserved	Reserved	RFST	TFIE	RFIE	RXTXCNT	CNFG
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	**Be sure to write "000".**			Bytes used in RX FIFO 0: Maximum 1:Same as FILL level of RX FIFO	TX interrupt for TX FIFO 0: Disabled 1: Enabled	RX interrupt for RX FIFO 0:Disabled 1: Enabled	Automatic disable of RXE/TXE 0:None 1:Auto disable	FIFO enable 0: Disabled 1: Enabled

<RFST>: When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected **(see Note 1)**.
0: The maximum number of bytes of the FIFO configured (see also <CNFG>).
1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL1:0>.

<TFIE>: When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFIE>: When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<RXTXCNT>: Controls automatic disabling of transmission and reception.
The mode control register SC0MOD1 <FDPX1:0> is used to set the types of TX/RX.
Setting "1" enables to operate as follows.

Half duplex RX	When receive shift register, the receive buffer and the RX FIFO are filled,SC0MOD0<RXE> is automatically set to "0" to inhibit further reception.
Half duplex TX	When the TX FIFO, the transmit buffer and the transmit shift register is empty,SC0MOD1<TXE> is automatically set to "0" to inhibit further transmission.
Full duplex	When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<CNFG>: Enables FIFO.
If enabled, the SCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows: (The type of TX/RX can be specified in the mode control register 1 SC0MOD1<FDPX1:0>).

Half duplex RX	RX FIFO 4byte
Half duplex TX	TX FIFO 4byte
Full duplex	RX FIFO 2byte＋TX FIFO 2byte

(Note 1) Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.

(Note 2) The FIFO can not use in 9bit UART mode.

13.4.9 RX FIFO configuration register

SC0RFC

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS					RIL1	RIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0				0	0
Function	RX FIFO clear 1: Clear “0” is read.	Select interrupt generation condition 0: when the data reaches to the specified fill level. 1: when the data reaches to the specified fill level or the data exceeds the specified fill level at the time data is read.	“0” is read.				FIFO fill level to generate RX interrupts 00:4byte 01:1byte 10:2byte 11:3byte	

<RFCS>: Clears RX FIFO
Setting “1” clears RX FIFO and “0” is always read.

<RFIS>: Specifies the condition of interrupt generation.
0: An interrupt is generated when it reaches to the specified fill level.
An interrupt is generated when it is reaches to the specified fill level or if it exceeds the specified fill level at the time data is read.

<RIL1:0>: Specifies FIFO fill level.

	Other than full duplex	Full duplex
00	4byte	2byte
01	1byte	1byte
10	2byte	2byte
11	3byte	1byte

(Note) To use TX/RX FIFO buffer, TX/RX FIFO must be cleared after setting the SIO transfer mode (half duplex/full duplex) and enabling FIFO (SC0FCNF<CNFG> = "1").

13.4.10 TX FIFO configuration register

SC0TFC

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS					TIL1	TIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0				0	0
Function	TX FIFO clear 1:Clear Always reads “0”.	Select interrupt generation condition 0: when the data reaches to the specified fill level. 1: when the data reaches to the specified fill level or the data cannot reach the specified fill level at the time new data is read.	“0” is read.				FIFO fill level to generate TX interrupts. 00:Empty 01:1byte 10:2byte 11:3byte	

<TFCS>: Clears TX FIFO.
Setting “1” clears TX FIFO and “0” is always read.

<TFIS>: Selects interrupt generation condition.
0: An interrupt is generated when the data reaches to the specified fill level.
1: An interrupt is generated when the data reaches to the specified fill level or the data cannot reach the specified fill level at the time new data is read.

<TIL1:0>: Selects FIFO fill level.

	Other than full duplex	Full duplex
00	Empty	Empty
01	1byte	1byte
10	2byte	Empty
11	3byte	1byte

(Note 1) To use TX/RX FIFO buffer, TX/RX FIFO must be cleared after setting the SIO transfer mode (half duplex/full duplex) and enabling FIFO (SC0FCNF<CNFG> = "1").

(Note 2) Set the SC0TFC register again After the following operation
SC0EN<SIOE> = "0" (SIO operation stop)
SC0MOD1<I2SC> = "0" (operation is prohibited in IDLE mode) and releasing the low power consumption mode which started by the WFI (Wait For Interrupt) instruction.

13.4.11 RX FIFO status register

SC0RST

	7	6	5	4	3	2	1	0
bit Symbol	ROR					RLVL2	RLVL1	RLVL0
Read/Write	R	R				R		
After reset	0	0				0	0	0
Function	RX FIFO Overrun 1: Generated	“0” is read.				Status of RX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

<ROR>: Flags for RX FIFO overrun. When the overrun occurs, these bits are set to “1” **(see note).**

<RLVL2:0>: Shows the fill level of RX FIFO.

(Note) The <ROR> bit is cleared to “0” when receive data is read from the SC0BUF register.

13.4.12 TX FIFO status register

SC0TST

	7	6	5	4	3	2	1	0
bit Symbol	TUR					TLVL2	TLVL1	TLVL0
Read/Write	R	R				R		
After reset	1	0				0	0	0
Function	TX FIFO Under run 1:Generated Cleared by writing FIFO	“0” is read.				Status of TX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

<TUR>: Flags for TX FIFO underrun. When the underrun occurs, these bits are set to “1” **(see note).**

<TLVL2:0>: Shows the fill level of TX FIFO.

(Note) The <TUR> bit is cleared to “0” when transmit data is written to the SC0BUF register.

13.5 Operation in Each Mode

13.5.1 Mode 0 (I/O interface mode)

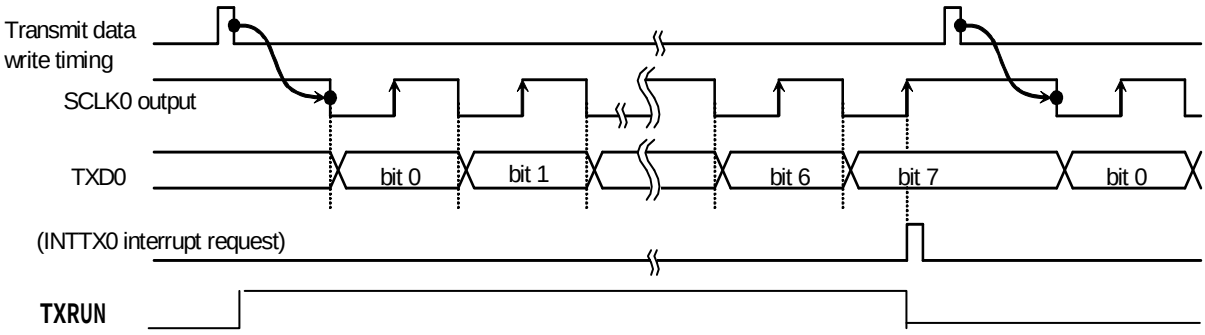
Mode 0 consists of two modes, the “SCLK output” mode to output synchronous clock and the “SCLK input” mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections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

① Transmitting data

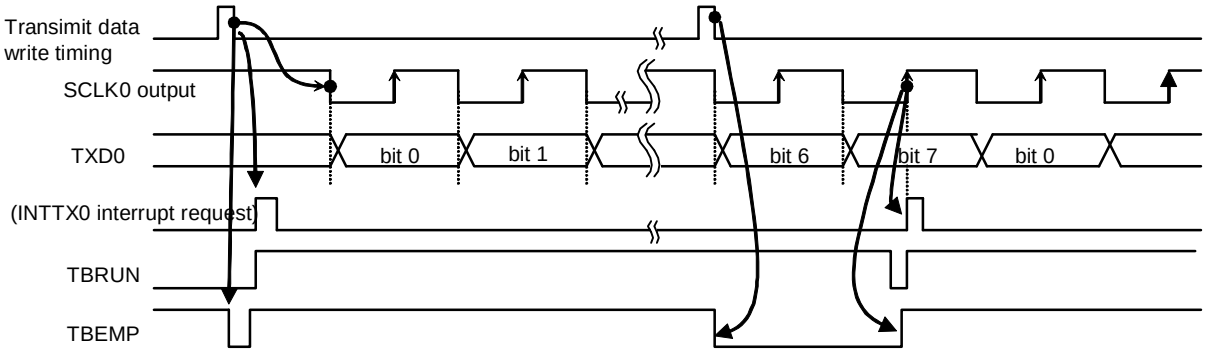
SCLK output mode

In the SCLK output mode, if SC0MOD2<WBUF> is set to “0” and the transmit double buffers are disabled, 8 bits of data are output from the TXD0 pin and the synchronous clock is output from the SCLK0 pin each time the CPU writes data to the transmit buffer. When all data is output, the INTTX0 interrupt is generated.

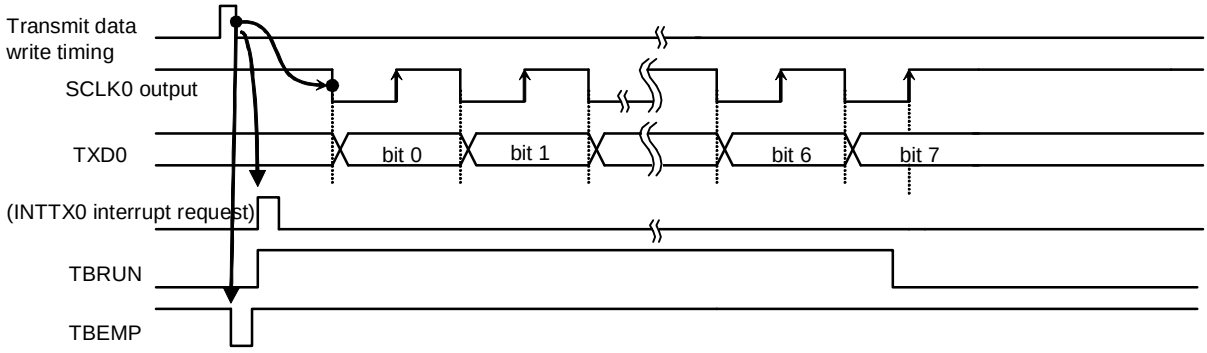
If SC0MOD2 <WBUF> is set to “1” and the transmit double buffers are enabled, data is moved from Transmit Buffer to Transmit shift register when the CPU writes data to Transmit Buffer while data transmission is halted or when data transmission from Transmit shift register (shift register) is completed. When data is moved from Transmit Buffer to Transmit shift register, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to “1,” and the INTTX0 interrupt is generated. If Transmit Buffer has no data to be moved to Transmit shift register , the INTTX0 interrupt is not generated and the SCLK0 output stops.



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



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



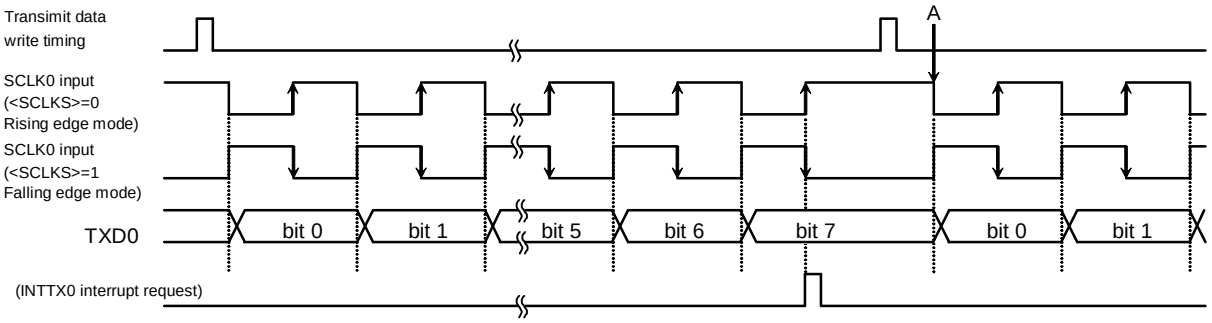
<WBUF>="1" (if double buffering is enabled and there is no data in buffer)

Fig 13-10 Transmit Operation in the I/O Interface Mode (SCLK0 Output Mode)

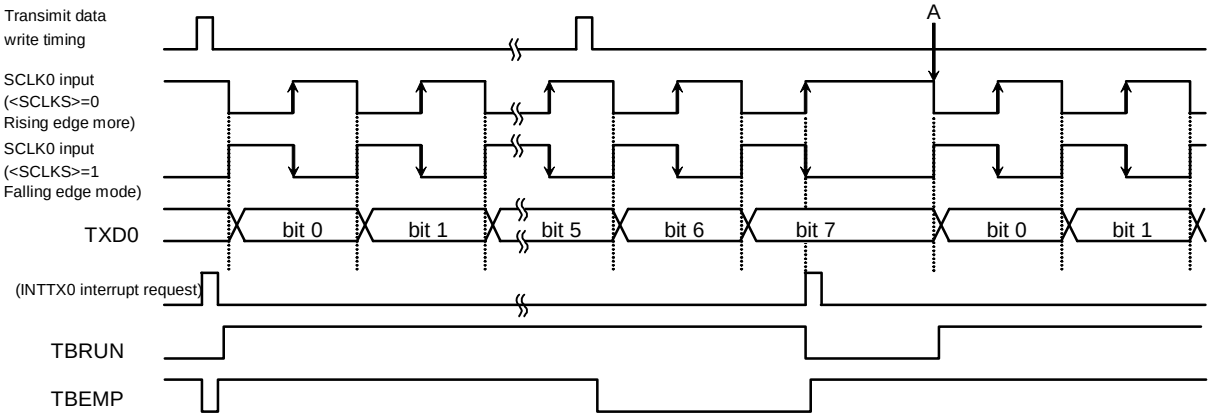
SCLK input mode

In the SCLK input mode, if SC0MOD2 <WBUF> is set to “0” and the transmit double buffers are disabled, 8-bit data that has been written in the transmit buffer is output from the TXD0 pin when the SCLK0 input becomes active. When all 8 bits are sent, the INTTX0 interrupt is generated. The next transmit data must be written before the timing point “A” as shown in Fig 13-11.

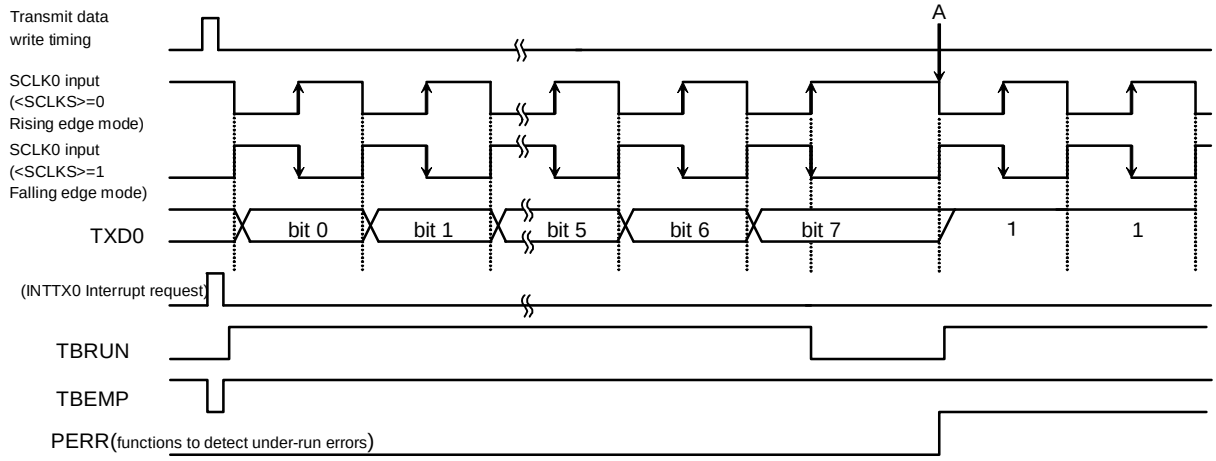
If SC0MOD2 <WBUF> is set to “1” and the transmit double buffers are enabled, data is moved from Transmit Buffer to Transmit shift register when the CPU writes data to Transmit Buffer before the SCLK0 becomes active or when data transmission from Transmit shift register is completed. As data is moved from Transmit Buffer to Transmit shift register, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to “1” and the INTTX0 interrupt is generated. If the SCLK0 input becomes active while no data is in Transmit Buffer, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is sent.



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



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



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

Fig 13-11 Transmit Operation in the I/O Interface Mode (SCLK0 Input Mode)

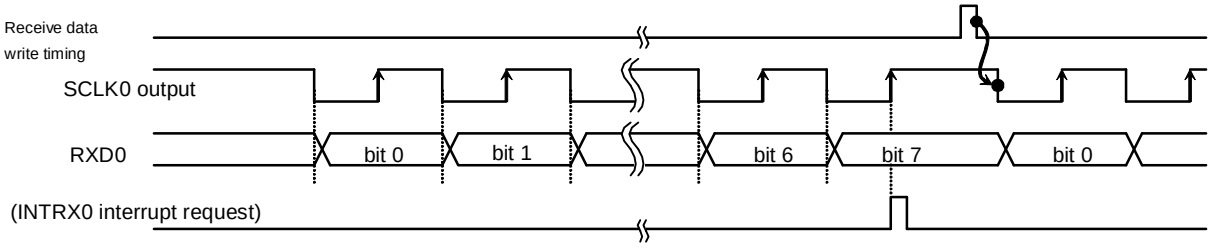
② Receiving data

SCLK output mode

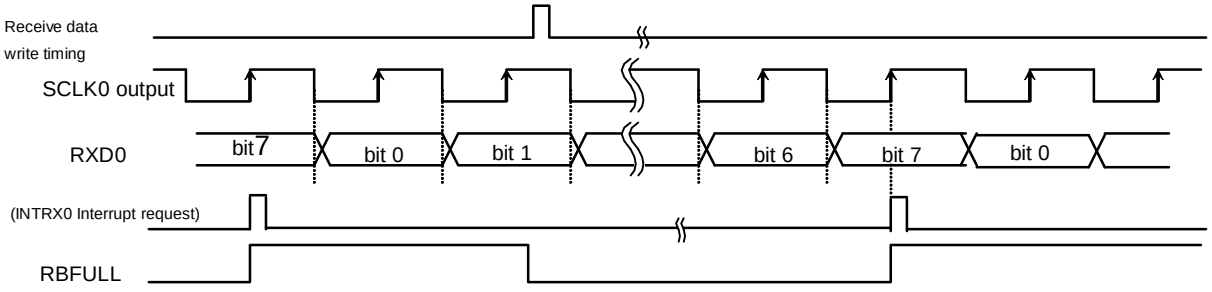
In the SCLK output mode, if SC0MOD2 <WBUF> = “0” and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive shift register each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

The first SCLK output can be started by setting the receive enable bit SC0MOD0 <RXE> to “1.” If the receive double buffering is enabled with SC0MOD2 <WBUF> set to “1,” the first frame received is moved to receive buffer and receive shift register can receive the next frame successively. As data is moved from receive shift register to receive buffer, the receive buffer full flag SC0MOD2 <RBFULL> is set to “1” and the INTRX0 interrupt is generated.

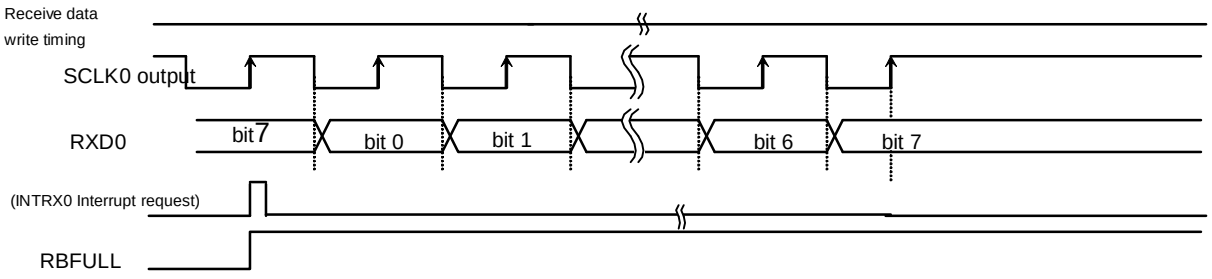
While data is in receive buffer, if CPU/DMAC cannot read data from receive buffer before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer allows data in receive shift register to move to receive buffer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



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



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



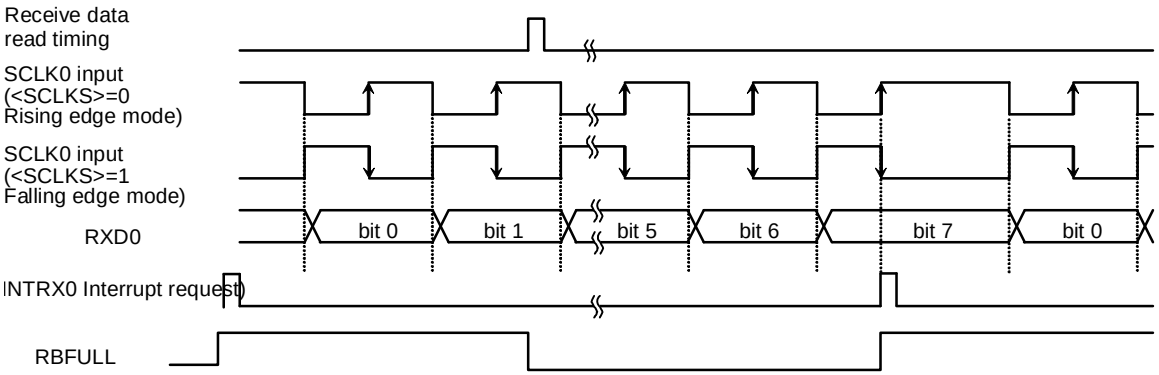
<WBUF>=“1” (if double buffering is enabled and data cannot be read from buffer)

Fig 13-12 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

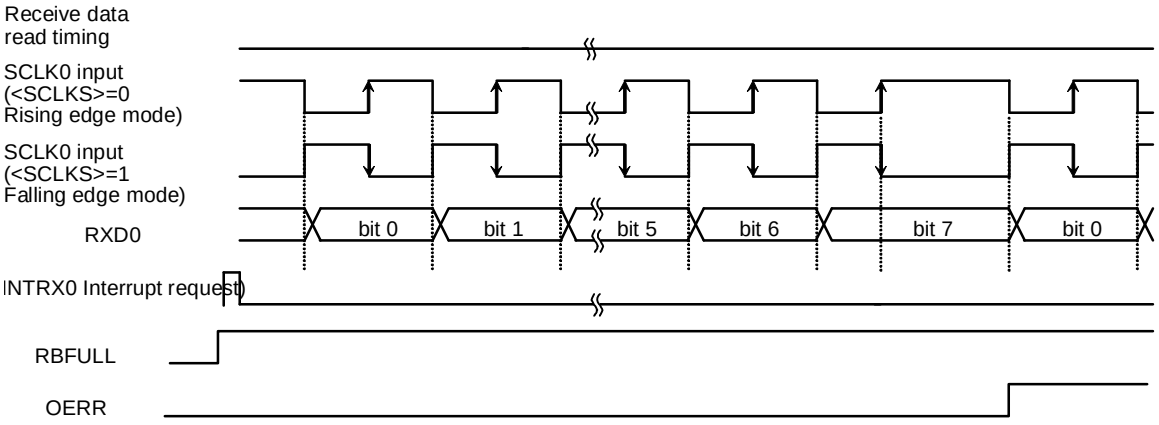
SCLK input mode

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

The INTRX0 receive interrupt is generated each time received data is moved to received buffer.



If data is read from buffer



If data cannot be read from buffer

Fig 13-13 Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

(Note) To receive data, SC0MOD <RXE> must always be set to "1" (receive enable) in the SCLK output / SCLK input mode.

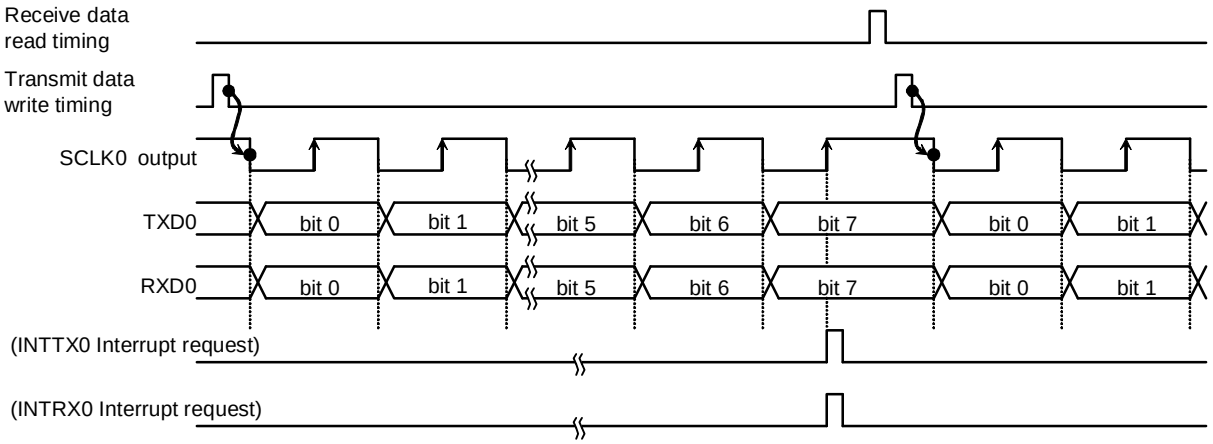
③ Transmit and receive (full-duplex)

The full-duplex mode is enabled by setting bit 6 <FDPX0> of the serial mode control register 1 (SC0MOD1) to “1”.

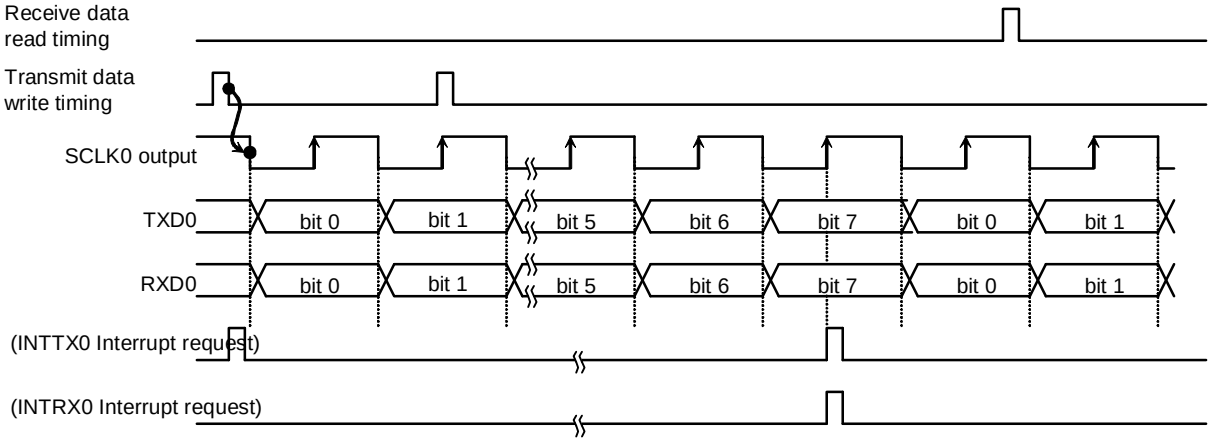
SCLK output mode

In the SCLK output mode, if SC0MOD2 <WBUF> is set to “0” and both the transmit and receive double buffers are disabled, SCLK is output when the CPU writes data to the transmit buffer. Subsequently, 8 bits of data are shifted into receive shift register and the INTRX0 receive interrupt is generated. Concurrently, 8 bits of data written to the transmit buffer are output from the TXD0 pin, the INTTX0 transmit interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops. In this, the next round of data transmission and reception starts when the data is read from the receive buffer and the next transmit data is written to the transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

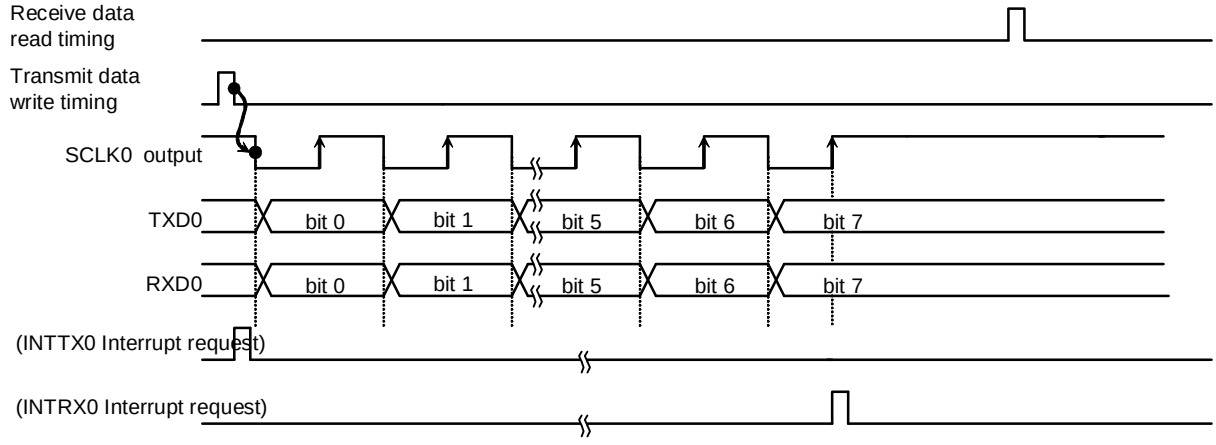
If SC0MOD2 <WBUF> = “1” and double buffering is enabled for both transmission and reception, SCLK is output when the CPU writes data to the transmit buffer. Subsequently, 8 bits of data are shifted into receive shift register, moved to receive buffer, and the INTRX0 interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is output from the TXD0 pin. When all data bits are sent out, the INTTX0 interrupt is generated and the next data is moved from the Transmit Buffer to Transmit shift register. If Transmit Buffer has no data to be moved to Transmit shift register (SC0MOD2 <TBEMP> = 1) or when receive buffer is full (SC0MOD2 <RBFULL> = 1), the SCLK clock is stopped. When both conditions, receive data is read and transmit data is written, are satisfied, the SCLK output is resumed and the next round of data transmission is started.



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



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



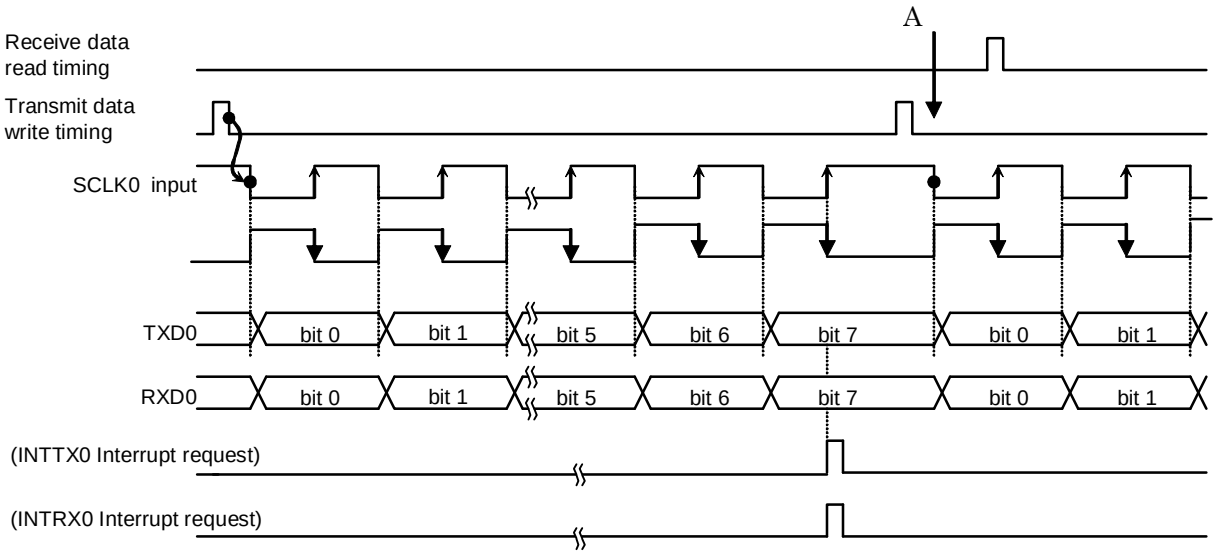
<WBUF>="1" (if double buffering is enabled)

Fig 13-14 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

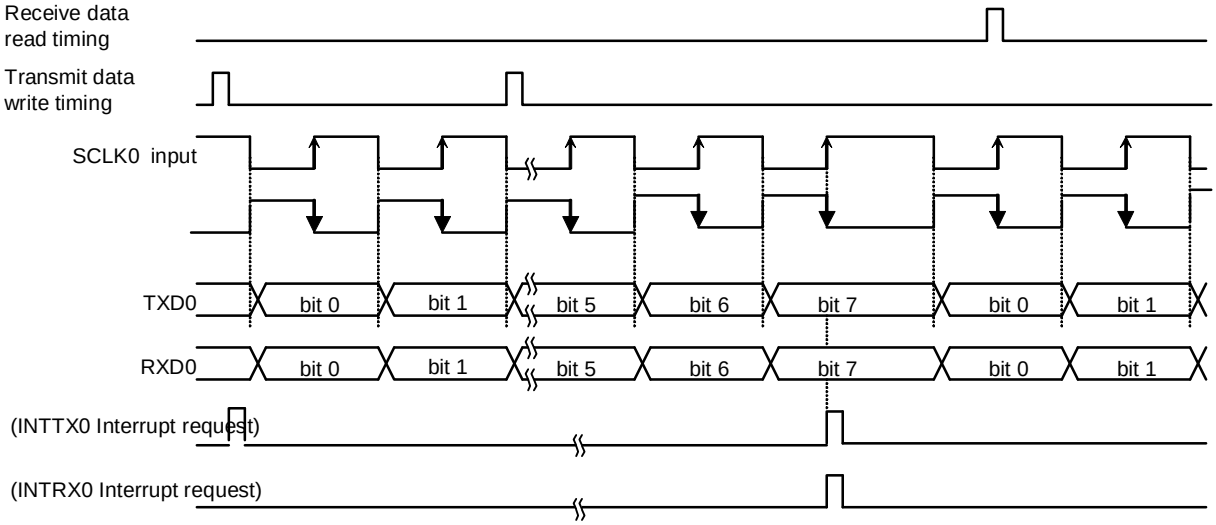
SCLK input mode

In the SCLK input mode with SC0MOD2 <WBUF> set to “0” and the transmit double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the transmit buffer is output from the TXD0 pin and 8 bits of data is shifted into the receive buffer when the SCLK input becomes active. The INTTX0 interrupt is generated upon completion of data transmission and the INTRX0 interrupt is generated at the instant the received data is moved from receive shift register to receive buffer. Note that transmit data must be written into the transmit buffer before the SCLK input for the next frame (data must be written before the point A in Fig 13-15). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

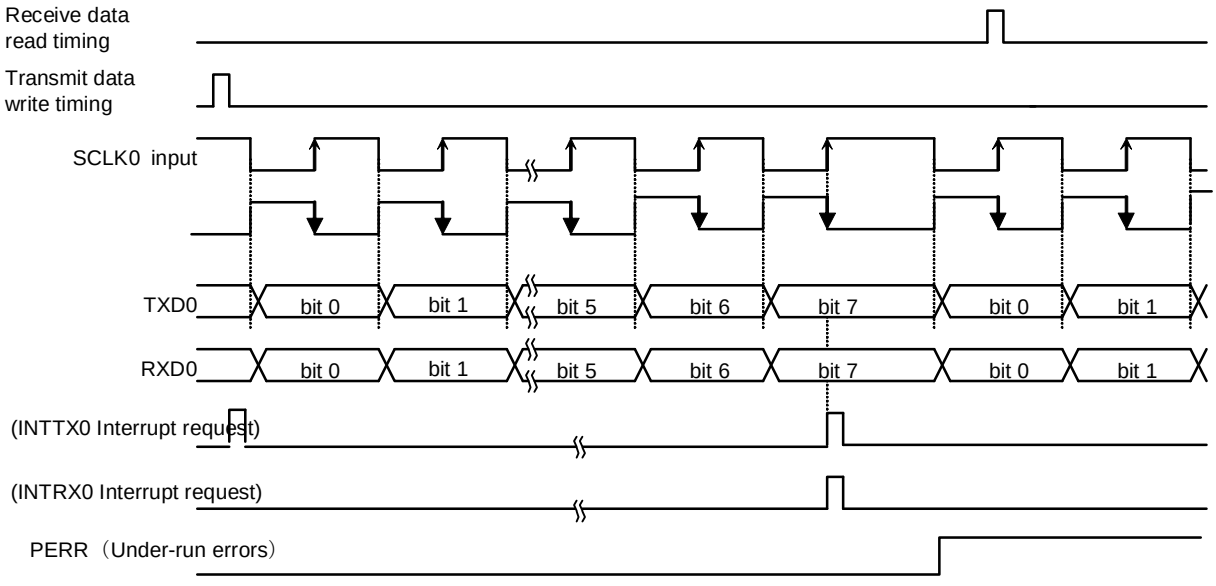
If SC0MOD2 <WBUF> = “1” and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing Transmit Buffer data is moved to Transmit shift register after completing data transmission from Transmit shift register. At the same time, the 8 bits of data received is shifted to shift register, it is moved to receive buffer, and the INTRX0 interrupt is generated. Upon the SCLK input for the next frame, transmission from Transmit shift register (in which data has been moved from Transmit Buffer) is started while receive data is shifted into receive shift register simultaneously. If data in receive buffer has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to Transmit Buffer when SCLK for the next frame is input, an under-run error occurs.



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



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



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

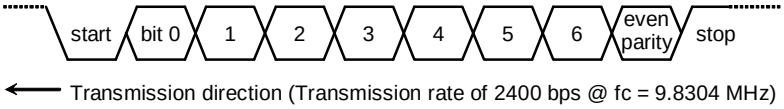
Fig 13-15 Transmit/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

13.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SC0MOD <SM1, 0>) to “01”.

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to “1” (enable), either even or odd parity may be selected using the SC0CR <EVEN> bit. The length of the stop bit can be specified using SC0MOD2<SBLEN>.

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



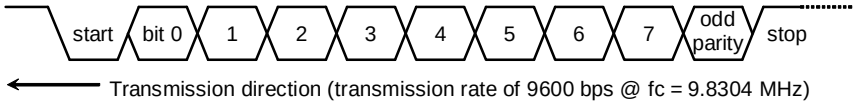
← Transmission direction (Transmission rate of 2400 bps @ fc = 9.8304 MHz)

* Clocking conditions	System clock	:	high- speed (fc)
	High-speed clock gear	:	x1 (fc)
	Prescaler clock	:	$f_{periph/2}$ ($f_{periph = fsys}$)

13.5.3 Mode 2 (8-bit UART Mode)

The 8-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to “10.” In this mode, parity bits can be added and parity enable/disable is controlled using SC0CR <PE>. If <PE> = “1” (enabled), either even or odd parity can be selected using SC0CR <EVEN>.

The control register settings for receiving data in the following format are as follows:



Transmission direction (transmission rate of 9600 bps @ fc = 9.8304 MHz)

* Clocking conditions	System clock	:	High-speed (fc)
	High-speed clock gear	:	x1 (fc)
	Prescaler clock	:	$f_{periph/4}$ ($f_{periph = fsys}$)

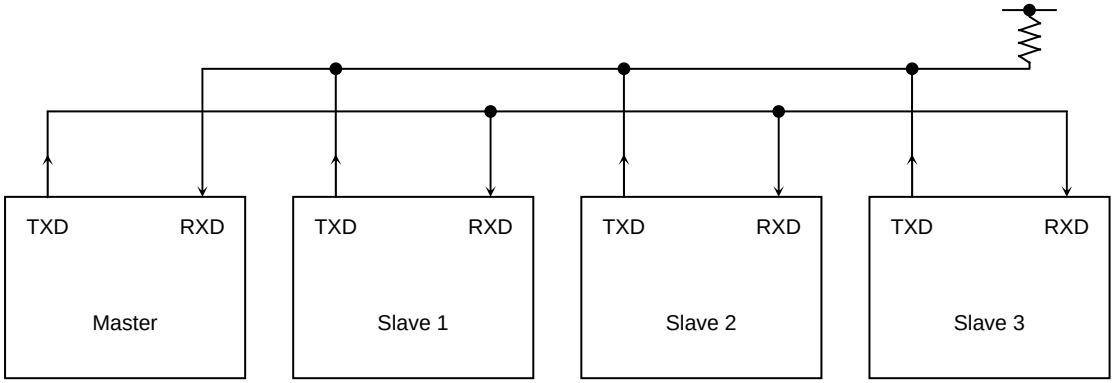
13.5.4 Mode 3 (9-bit UART)

The 9-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to “11.” In this mode, parity bits must be disabled (SC0CR <PE> = “0”).

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SC0MOD0) for transmitting data. The data is stored in bit 7 <RB8> of the serial control register SC0CR. When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SC0BUF. The stop bit length can be specified using SC0MOD2 <SBLEN>.

Wakeup function

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



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

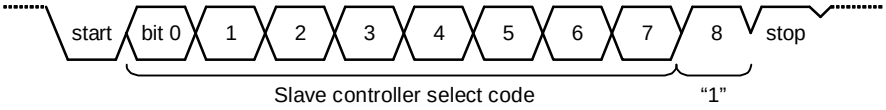
Fig 13-16 Serial Links to Use Wake-up Function

Protocol

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

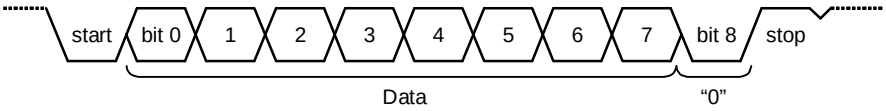
② Set SC0MOD <WU> to “1” for the slave controllers to make them ready to receive data.

③ The master controller is to transmit a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to “1”.



④ Each slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to “0”.

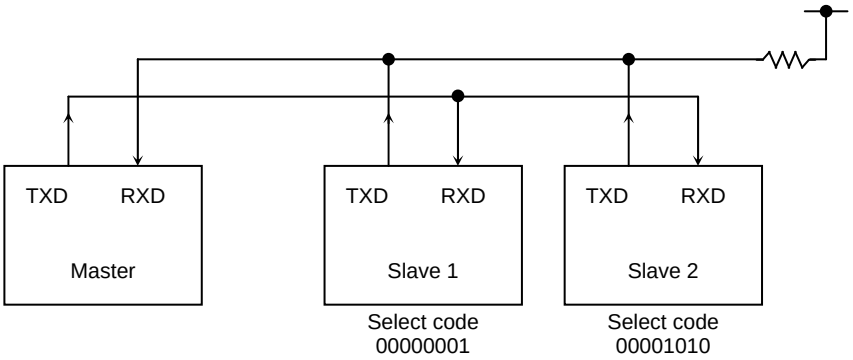
⑤ The master controller transmits data to the designated slave controller (the controller of which SC0MOD <WU> bit is cleared to “0”). In this, the most significant bit (bit 8) <TB8> must be set to “0”.



⑥ The slave controllers with the <WU> bit set to “1” ignore the receive data because the most significant bit (bit 8) <RB8> is set to “0” and thus no interrupt (INTRX0) is generated.

Also, the slave controller with the <WU> bit set to “0” can transmit data to the master controller to inform that the data has been successfully received.

An example: Using the internal clock f_{SYS} as the transfer clock, two slave controllers are serially linked as follows.



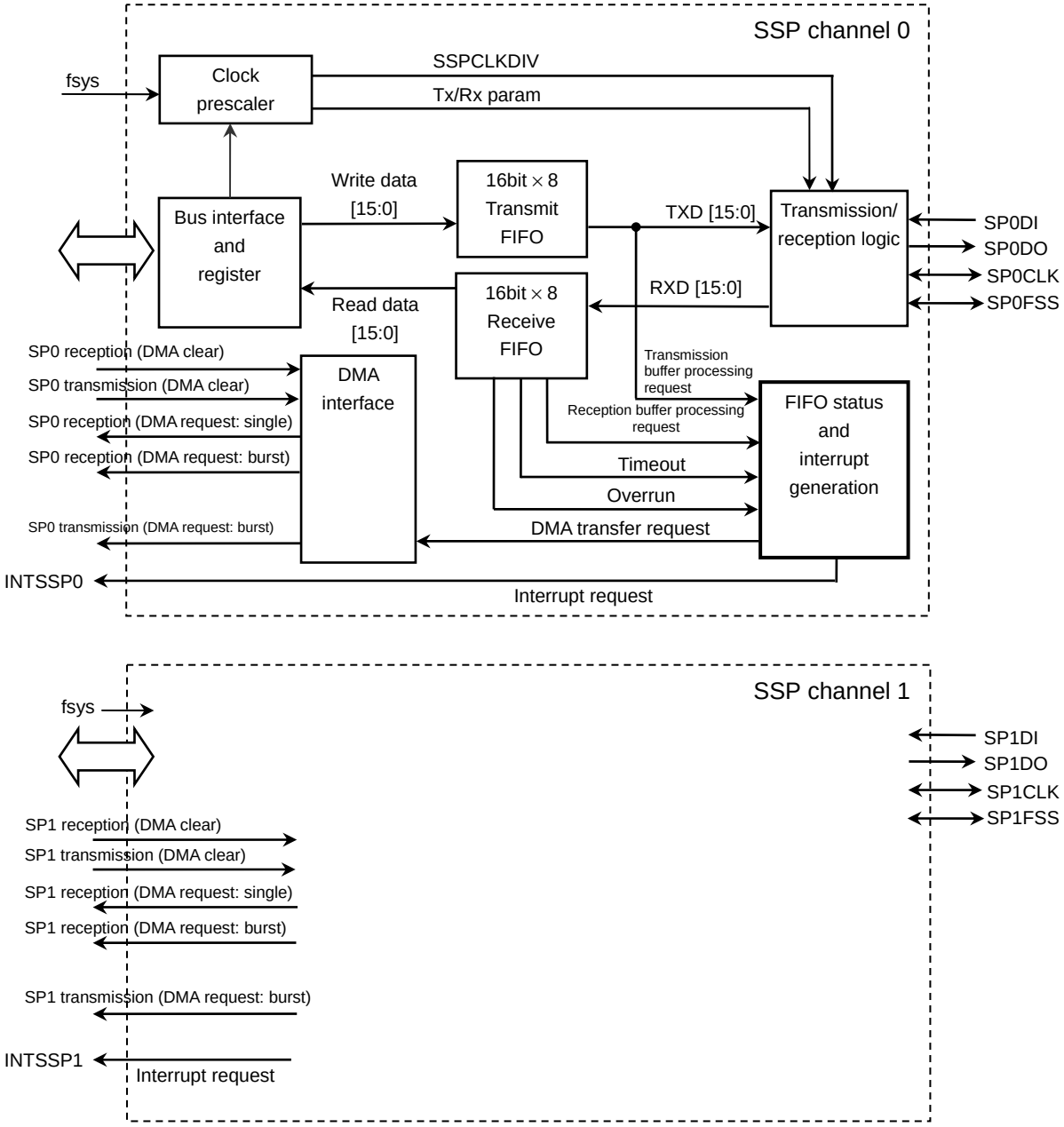
14. SSP (Synchronous Serial Port)

Important
TMPM382 (64-pin version) does not implement SSP1.
Please do not use these functions if you use this product.

This LSI contains an SSP (Synchronous Serial Port) with two channels.
Each channel has the following features:

	Channels 0 to 1	
Communication protocol	Three types of synchronous serial ports including the SPI	
Operation mode	Master/slave mode	
Transmit FIFO	16 bits wide / 8 tiers deep	
Receive FIFO	16 bits wide / 8 tiers deep	
transmitted/ received data size	4 to 16 bits	
Interrupt type	Transmit interrupt Receive interrupt Receive overrun interrupt Timeout interrupt	
Communication speed	In master mode: fsys / 2 (max. 10Mbps: When System clock is 40MHz, divide by 4)	
	In slave mode: fsys (40MHz) / 12 (max. 3.3Mbps) (When slave mode is selected, SSPxCR0<SCR>=0x00, SSPxCPSR=0x2)	
DMA	Supported	
Internal test function	Can use the internal loopback test mode.	
Control pin	Channel 0	Channel 1
	SP0CLK SP0FSS SP0DO SP0DI	SP1CLK SP1FSS SP1DO SP1DI

14.1 Block Diagram



14.2 Overview of SSP

This LSI contains the SSP with two channels: channels 0, and 1 channels operate in the same way, only channel 0 is described in the following sections.

The SSP is an interface that enables serial communications with the peripheral devices with three types of synchronous serial interface functions.
The SSP performs serial-parallel conversion of the data received from a peripheral device. The transmit path buffers data in the independent 16-bit wide and 8-layered transmit FIFO in the transmit mode, and the receive path buffers data in the 16-bit wide and 8-layered receive FIFO in receive mode. Serial data is transmitted via SP0DO and received via SP0DI.
The SSP contains a programmable prescaler to generate the serial output clock SP0CLK from the input clock fsys. The operation mode, frame format, and data size of the SSP are programmed in the control registers SSP0CR0 and SSP0CR1.

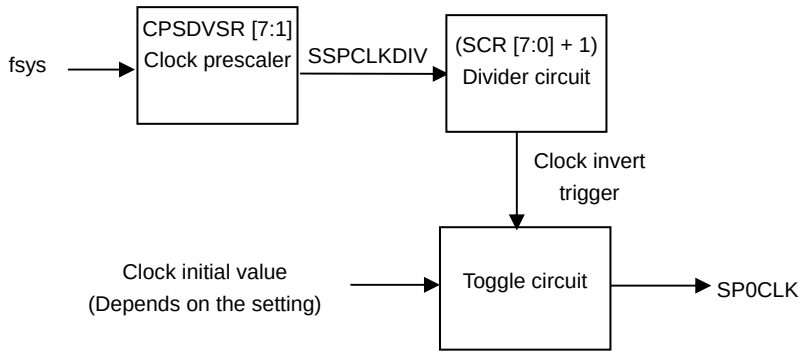
(1) Clock prescaler

When configured as a master, a clock prescaler comprising two free-running serially linked counters is used to provide the serial output clock SP0CLK.

You can program the clock prescaler through the SSP0CPSR register, to divide fsys by a factor of 2 to 254 in steps of two. Because the least significant bit of the SSP0CPSR register is not used, division by an odd number is not possible.

The output of the prescaler is further divided by a factor of 1 to 256, which is obtained by adding 1 to the value programmed in the SSP0CR0 control register, to give the master output clock SP0CLK.

$$\text{Bit rate} = \text{fsys} / (\text{CPSDVSR} \times (1+\text{SCR}))$$



(2) Transmit FIFO

This is a 16-bit wide, 8-layered transmit FIFO buffer, which is shared in master and slave modes.

(3) Receive FIFO

This is a 16-bit wide 8-layered receive FIFO buffer, which is shared in master and slave modes.

(4) Interrupt generation logic

Four HIGH active interrupts, each of which can be masked separately, are generated. Also, individual interrupt requests are combined and output as a single integrated interrupt.

- Transmit interrupt: Interrupt conditional upon TxFIFO having free space equal to or more than half its entire capacity.
  (Number of valid data items in the TxFIFO ≤ 4)
- Receive interrupt: Interrupt conditional upon RxFIFO having valid data equal to or more than half its entire capacity.
  (Number of valid data items in the RxFIFO ≥ 4)
- Timeout interrupt: Interrupts indicating that the data in RxFIFO is not read before the timeout period expires.
- Receive overrun interrupt: Conditional interrupts indicating that data is written to RxFIFO when it is full

When any of the above interrupts is asserted, INTSSP0 is asserted.

(a) Transmit interrupt

The transmit interrupt is asserted when there are four or fewer valid entries in the transmit FIFO. The transmit interrupt is also generated when the SSP operation is disabled (SSP0CR1<SSE>=0).

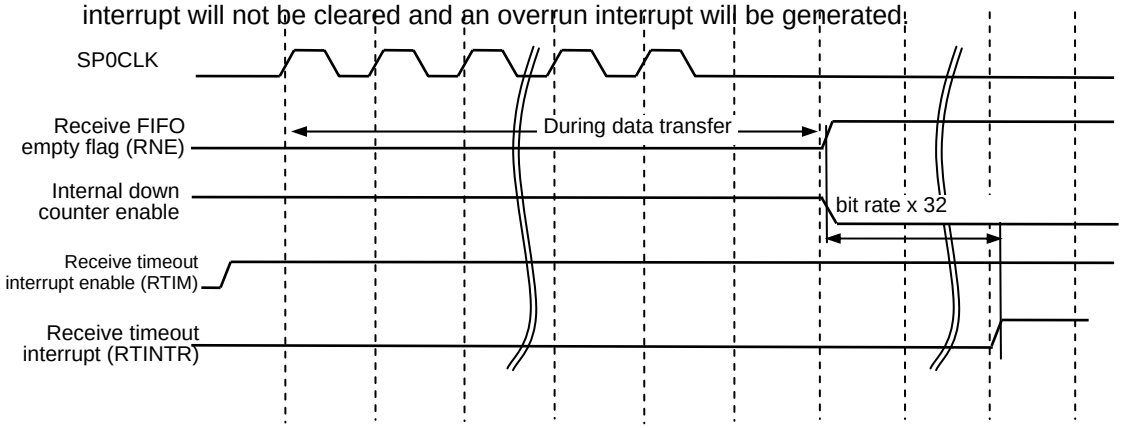
The first transmitted data can be written in the FIFO by using this interrupt.

(b) Receive interrupt

The receive interrupt is asserted when there are four or more valid entries in the receive FIFO.

(c) Timeout interrupt

The receive timeout interrupt is asserted when the receive FIFO is not empty and the SSP has remained idle for a fixed 32-bit period (bit rate). This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing. This operation occurs in both master and slave modes. When the timeout interrupt is generated, read all data from the receive FIFO. Even if all the data is not read, data can be transmitted/received if the receive FIFO has a free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. When transfer starts, the timeout interrupt will be cleared. If data is transmitted/received when the receive FIFO has no free space, the timeout interrupt will not be cleared and an overrun interrupt will be generated.

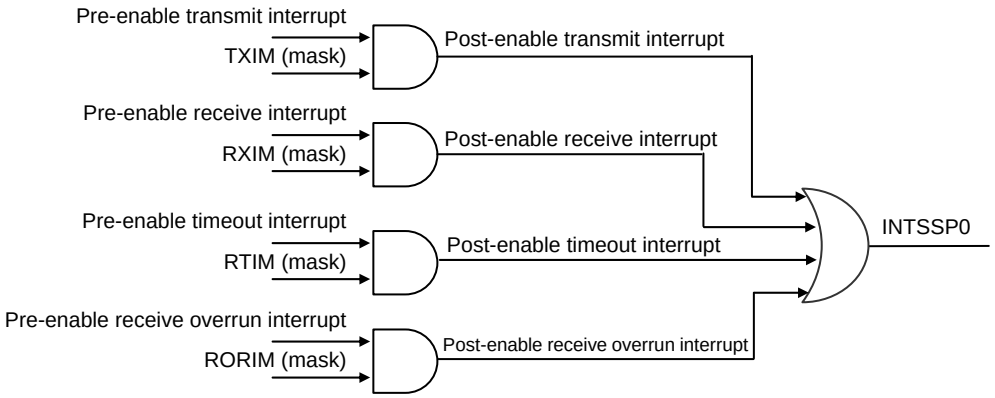


(d) Receive overrun interrupt

When the next data (9th data item) is received when the receive FIFO is already full, a receive overrun interrupt is generated immediately after transfer. The data received after the receive overrun interrupt is generated (including the 9th data item) will become invalid and be discarded. However, if data is read from the receive FIFO while the 9th data item is being received (before the interrupt is generated), the 9th received data will be written in the receive FIFO as valid data. To perform transfer properly when the receive overrun interrupt has been generated, write "1" to the receive overrun interrupt clear register, and then read all data from the receive FIFO. Even if all the data is not read, data can be transmitted/received if the receive FIFO has free space and the number of data to be transmitted does not exceed the free space of the receive FIFO. Note that if the receive FIFO is not read (provided that the receive FIFO is not empty) within a certain 32-bit period (bit rate) after the receive overrun interrupt is cleared, a timeout interrupt will be generated.

(e) Combined interrupt

The above four interrupts combine individual masked sources into a single interrupt. When any of the above interrupts is asserted, the integrated interrupt INTSSP0 is asserted.



(5) DMA interface

The SSP provides an interface to connect to a DMA controller.

14.3 SSP Operation

(1) Initial settings for SSP

Settings for the SSP communication protocol must be made with the SSP disabled.

Control registers SSP0CR0 and SSP0CR1 need to configure this SSP as a master or slave operating under one of the following protocols. In addition, make the settings related to the communication speed in the prescale registers SSP0CPSR and SSP0CR0<SCR>.

This SSP supports the following protocols:

- SPI, SSI, Microwire

(2) SSP enable

The transfer operation starts when the operation is enabled with the transmitted data written in the transmit FIFO, or when transmitted data is written in the transmit FIFO with the operation enabled.

However, if the transmit FIFO contains only four or fewer entries when the operation is enabled, a transmit interrupt will be generated. This interrupt can be used to write the initial data.

Note) When the SSP is in the SPI slave mode and the FSS pin is not used, be sure to transmit data of one byte or more in the FIFO before enabling the operation. If the operation is enabled with the transmit FIFO empty, the transfer data will not be output correctly.

(3) Clock ratios

When setting a frequency for PCLK, the following conditions must be met.

[In master mode]

f_{SP0CLK} (maximum) => fsys / 2 (Note)

f_{SP0CLK} (minimum) => fsys / (254 x 256)

[In slave mode]

f_{SP0CLK} (maximum) => fsys / 12

f_{SP0CLK} (minimum) => fsys / (254 x 256)

Note) fsys is output from Clock Gear. In details, please refer the chapter of Clock Gear.
In TMPM380, maximum baud-rate is 10Mbps. When system clock is 40MHz, divide by 4)

(4) Frame format

Each frame format is between 4 and 16 bits wide depending on the size of data programmed, and is transmitted starting from the MSB.

- Serial clock (SP0CLK)

Signals remain LOW in the SSI and Microwire formats and as Inactive in the SPI format while the SSP is in the idle state. In addition, data is output at the set bit rate only during data transmission.

- Serial frame (SP0FSS)

In the SPI and Microwire frame formats, signals are set to LOW Active and always asserted to LOW during frame transmission.

In the SSI frame format, signals are asserted only during 1 bit rate before each frame transmission. In this frame format, output data is transmitted at the rising edge of SP0CLK and the input data is received at its falling edge.

- Notes on the Microwire

The Microwire format uses a special master/slave messaging method, which operates in half-duplex mode. In this mode, when a frame begins, an 8-bit control message is transmitted to the slave. During this transmit, no incoming data is received by the SSP. After the message has been transmitted, the slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, it responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

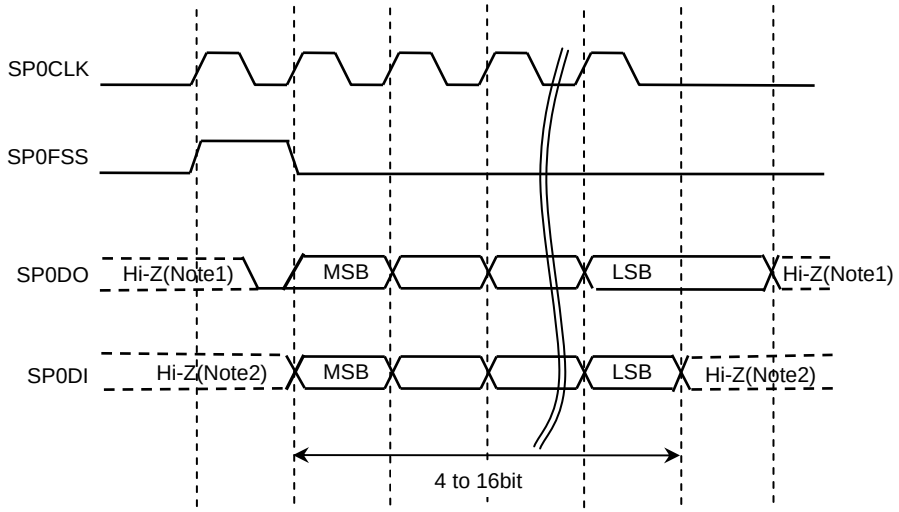
The details of each frame format are described below:

(a) SSI frame format

In this mode, the SSP is in idle state, SP0CLK and SP0FSS are forcedly set to LOW, and the transmit data line SP0DO becomes Hi-Z. When data is written in the transmit FIFO, the master outputs High pulses of 1 SP0CLK to the SP0FSS line. The transmitted data will be transferred from the transmit FIFO to the transmit serial shift register. Data of 4 to 16 bits will be output from the SP0DO pin at the next rising edge of SP0CLK.

Likewise, the received data will be input starting from the MSB to the SP0DI pin at the falling edge of SP0CLK. The received data will be transferred from the serial shift register into the receive FIFO at the rising edge of SP0CLK after its LSB data is latched.

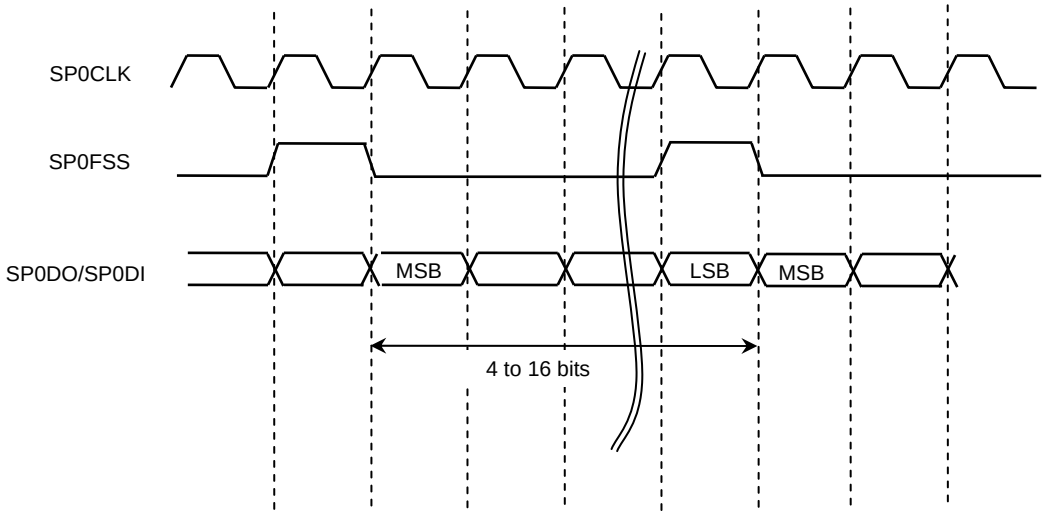
SSI frame format (transmission/reception during single transfer)



Note1) When transmission is disable , SP0DO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note2) SP0DI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

SSI frame format (transmission/reception during continuous transfer)



Note1) When transmission is disable , SP0DO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note2) SP0DI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

(b) SPI

The SPI interface has 4 lines. SP0FSS is used for slave selection. One of the main features of the SPI format is that the <SPO> and <SPH> bits in the SSP0CR0 control register can be used to set the SP0CLK operation timing.

SSP0CR0<SPO>

SSP0CR0<SPO> is used to set the level at which SP0CLK in idle state is held.

<SPO>=1: Sets SP0CLK in High state

<SPO>=0: Sets SP0CLK in Low state

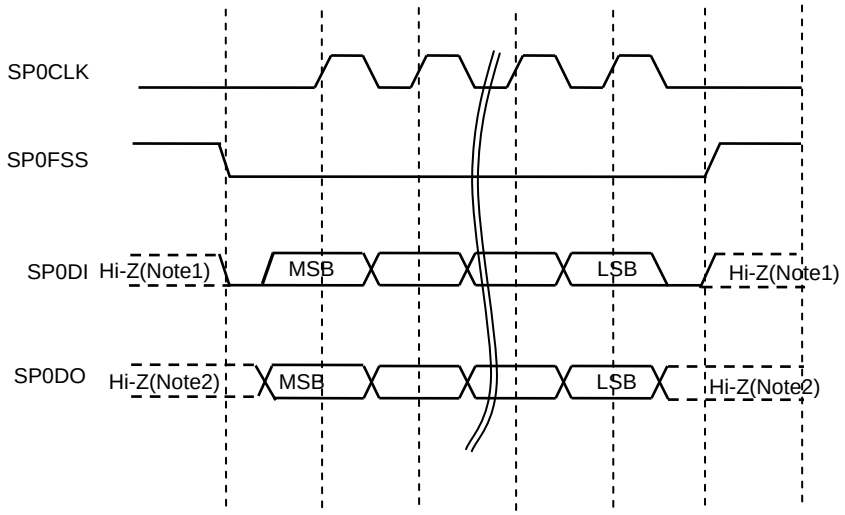
SSP0CR0<SPH>

SSP0CR0<SPH> is used to select the clock edge at which data is latched.

SSP0CR0<SPH>=0: Captures data at the 1st clock edge.

SSP0CR0<SPH>=1: Captures data at the 2nd clock edge.

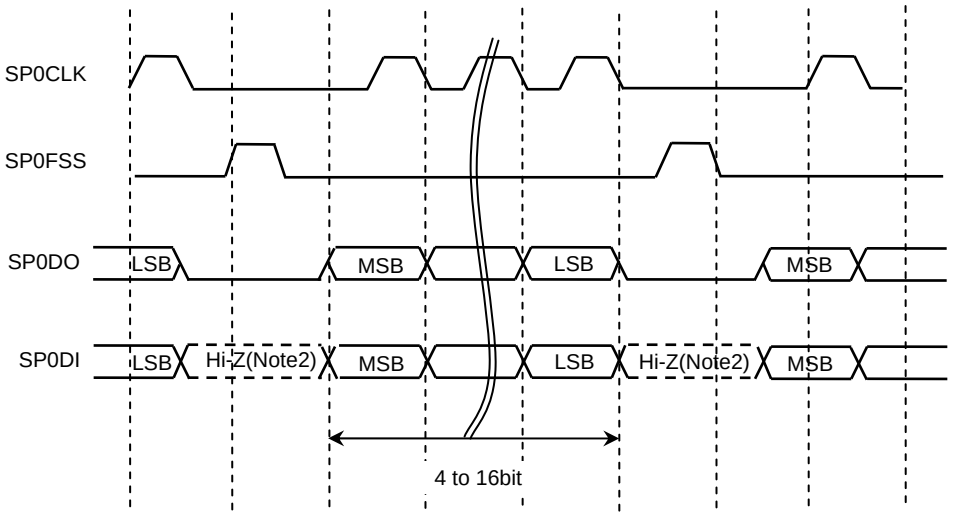
SPI frame format (single transfer, <SPO>=0 & <SPH>=0)



Note1) When transmission is disable , SP0DO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note2) SP0DI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

SPI frame format (continuous transfer, <SPO>=0 & <SPH>=0)



Note1) When transmission is disable , SP0DO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note2) SP0DI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

With this setting, during the idle period:

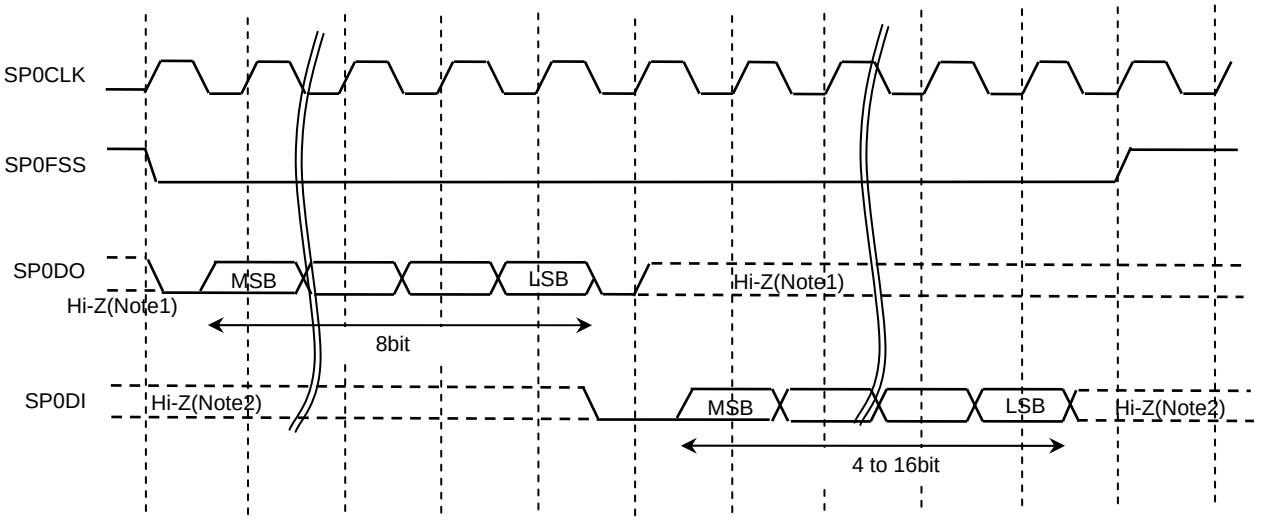
- The SP0CLK signal is forcedly set to LOW.
- SP0FSS is forcedly set to HIGH.
- The transmit data line SP0DO is set to LOW.

If the SSP is enabled and valid data exists in the transmit FIFO, the SP0FSS master signal driven by LOW notifies of the start of transmission. This enables the slave data in the SP0DI input line of the master.

When a half of the SP0CLK period has passed, valid master data is transferred to the SP0DO pin. Both the master data and slave data are now set. When another half of SP0CLK has passed, the SP0CLK master clock pin becomes HIGH. After that, the data is captured at the rising edge of the SP0CLK signal and transmitted at its falling edge. In the single word transfer, the SP0FSS line will return to the idle HIGH state when all the bits of that data word have been transferred, and then one cycle of SP0CLK has passed after the last bit was captured. However, for continuous transfer, the SP0FSS signal must be pulsed at HIGH between individual data word transfers. This is because change is not enabled when the slave selection pin freezes data in its peripheral register and the <SPH> bit is logical 0. Therefore, to enable writing of serial peripheral data, the master device must drive the SP0FSS pin of the slave device between individual data transfers. When the continuous transfer is complete, the SP0FSS pin will return to the idle state when one cycle of SP0CLK has passed after the last bit is captured.

(c) Microwire frame format

Microwire frame format (single transfer)



Note1) When transmission is disable , SP0DO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note2) SP0DI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Though the Microwire format is similar to the SPI format, it uses the master/slave message transmission method for half-duplex communications. Each serial transmission is started by an 8-bit control word, which is sent to the off-chip slave device. During this transmission, the SSP does not receive input data. After the message has been transmitted, the off-chip slave decodes it, and after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits. With this configuration, during the idle period:

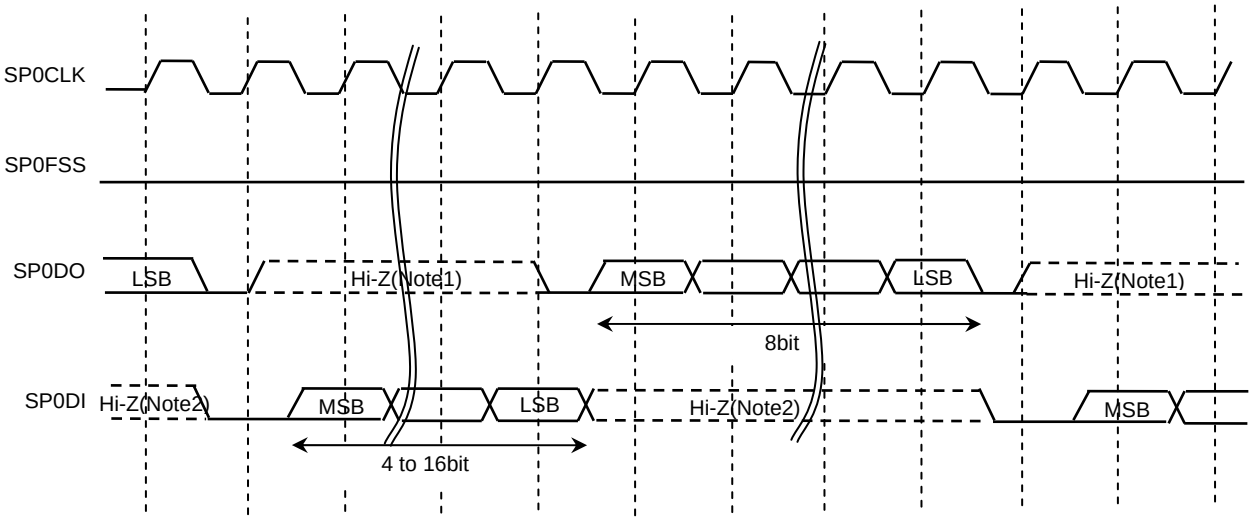
- The SP0CLK signal is forcedly set to LOW.
- SP0FSS is forcedly set to HIGH.
- The transmit data line SP0DO is set to LOW.

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SP0FSS causes the value stored in the bottom entry of the transmit FIFO to be transferred to the serial shift register for the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SP0DO pin. SP0FSS remains LOW and the SP0D1 pin remains tristated during this transmission. The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SP0CLK. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSP. Each bit is driven onto SP0DI line on the falling edge of SP0CLK. The SSP in turn latches each bit on the rising edge of SP0CLK. At the end of the frame, for single transfers, the SP0FSS signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter, which causes the data to be transferred to the receive

FIFO.

Note) The off-chip slave device can tristate the receive line either on the falling edge of SP0CLK after the LSB has been latched by the receive shifter, or when the SP0FSS pin goes HIGH.

Microwire frame format (continuous transfer)



Note1) When transmission is disable , SP0DO terminal doesn't output and is high impedance status. This terminal needs to add suitable pull-up/down resistance to valid the voltage level.

Note2) SP0DI terminal is always input and internal gate is open. In case of transmission signal will be high impedance status, this terminal needs to add suitable pull-up/down resistance to valid the voltage level.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SP0FSS line is continuously asserted (held LOW) and transmission of data occurs back to back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge of SP0CLK, after the LSB of the frame has been latched into the SSP.

Note) [Example of connection]

The SSP does not support dynamic switching between the master and slave in the system. Each sample SSP is configured and connected as either a master or slave.

(5) DMA interface

The DMA operation of the SSP is controlled through the DMA control register, SP0DMACR.

When there are more data than the watermark level (half of the FIFO) in the receive FIFO, the receive DMA request is asserted.

When the amount of data left in the receive FIFO is less than the watermark level (half of the FIFO), the transmit DMA request is asserted.

To clear the transmit/receive DMA request, an input pin for the transmit/receive DMA request clear signals, which are asserted by the DMA controller, is provided.

Set the DMA burst length to four words.

* For the remaining three characters, the SSP does not assert the burst request.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described above. All request signals are deasserted if the SSP is disabled or the DMA enable signal is cleared.

The following table shows the trigger points for DMABREQ, for both the transmit and receive FIFOs.

	Burst length	
Watermark level	Transmit (number of empty locations)	Receive (number of filled locations)
1/2	4	4

14.4 Explanation of the Register

The following lists the SFRs:

- SSP0

base address = 0x400C_0000

Register Name	Address (base+)	Description
SSP0CR0	0x0000	Control register 0
SSP0CR1	0x0004	Control register 1
SSP0DR	0x0008	Receive FIFO (read) and transmit FIFO data register (write)
SSP0SR	0x000C	Status register
SSP0CPSR	0x0010	Clock prescale register
SSP0IMSC	0x0014	Interrupt enable/disable register
SSP0RIS	0x0018	Pre-enable interrupt status register
SSP0MIS	0x001C	Post-enable interrupt status register
SSP0ICR	0x0020	Interrupt clear register
SSP0DMACR	0x0024	DMA control register
–	0x0028 ~ 0xFFC	Reserved

- SSP1

base address = 0x400C_1000

Register Name	Address (base+)	Description
SSP1CR0	0x0000	Control register 0
SSP1CR1	0x0004	Control register 1
SSP1DR	0x0008	Receive FIFO (read) and transmit FIFO data register (write)
SSP1SR	0x000C	Status register
SSP1CPSR	0x0010	Clock prescale register
SSP1IMSC	0x0014	Interrupt enable/disable register
SSP1RIS	0x0018	Pre-enable interrupt status register
SSP1MIS	0x001C	Post-enable interrupt status register
SSP1ICR	0x0020	Interrupt clear register
SSP1DMACR	0x0024	DMA control register
–	0x0028 ~ 0xFFC	Reserved

14.4.1 SSP0CR0 (SSP0 control register 0)

Address = (0x400C_0000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read undefined. Write as zero.
[15:8]	SCR	R/W	0y0	For serial clock rate setting Parameter: [Refer to Explanation]) 0x00 ~ 0xFF
[7]	SPH	R/W	0y0	SPCLK phase (applicable to Motorola SPI frame format only, Refer to [Motorola SPI frame format])
[6]	SPO	R/W	0y0	SPCLK polarity (applicable to Motorola SPI frame format only, Refer to [Motorola SPI frame format])
[5:4]	FRF	R/W	0y00	Frame format: 0y00: Motorola SPI frame format 0y01: TI synchronous Serial frame format 0y10: National Microwire Frame format 0y11: Reserved, undefined operation
[3:0]	DSS	R/W	0y0000	Data size select: 0y0000: Reserved, undefined operation 0y0001: Reserved, undefined operation 0y0010: Reserved, undefined operation 0y0011: 4-bit data 0y0100: 5-bit data 0y0101: 6-bit data 0y0110: 7-bit data 0y0111: 8-bit data 0y1000: 9-bit data 0y1001: 10-bit data 0y1010: 11-bit data 0y1011: 12-bit data 0y1100: 13-bit data 0y1101: 14-bit data 0y1110: 15-bit data 0y1111: 16-bit data

14.4.2 SSP1CR0 (SSP1 control register 0)

Address = (0x400C_1000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read undefined. Write as zero.
[15:8]	SCR	R/W	0y0	For serial clock rate setting Parameter: 0x00~0xFF
[7]	SPH	R/W	0y0	SPCLK phase (applicable to Motorola SPI frame format only, Refer to [Motorola SPI frame format])
[6]	SPO	R/W	0y0	SPCLK polarity (applicable to Motorola SPI frame format only, Refer to [Motorola SPI frame format])
[5:4]	FRF	R/W	0y00	Frame format: 0y00: Motorola SPI frame format 0y01: TI synchronous Serial frame format 0y10: National Microwire Frame format 0y11: Reserved, undefined operation
[3:0]	DSS	R/W	0y0000	Data size select: 0y0000: Reserved, undefined operation 0y0001: Reserved, undefined operation 0y0010: Reserved, undefined operation 0y0011: 4-bit data 0y0100: 5-bit data 0y0101: 6-bit data 0y0110: 7-bit data 0y0111: 8-bit data 0y1000: 9-bit data 0y1001: 10-bit data 0y1010: 11-bit data 0y1011: 12-bit data 0y1100: 13-bit data 0y1101: 14-bit data 0y1110: 15-bit data 0y1111: 16-bit data

[Explanation]

a. <SCR>

Used to generate the SPP transmit bit rate and receive bit rate.

This bit rate can be obtained by the following equation:
Bit rate = fsys / (CPSDVSR × (1+SCR))
CPSDVSR is an even number between 2 to 254, which is programmed by the SSPxCPSR register, and SCR takes a value between 0 to 255.

14.4.3 SSP0CR1 (SSP0 control register 1)

Address = (0x400C_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	SOD	R/W	0y0	Slave mode SP0DO output control: 0y0: Enable 0y1: Disable
[2]	MS	R/W	0y0	Master/slave mode select: 0y0: Device configured as a master 0y1: Device configured as a slave
[1]	SSE	R/W	0y0	SSP0 enable: 0y0: Disable 0y1: Enable
[0]	LBM	R/W	0y0	Loop back mode: 0y0: Normal serial port operation enabled 0y1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

14.4.4 SSP1CR1 (SSP1 control register 1)

Address = (0x400C_1000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	SOD	R/W	0y0	Slave mode SP0DO output control: 0y0: Enable 0y1: Disable
[2]	MS	R/W	0y0	Master/slave mode select: 0y0: Device configured as a master 0y1: Device configured as a slave
[1]	SSE	R/W	0y0	SSP1 enable: 0y0: Disable 0y1: Enable
[0]	LBM	R/W	0y0	Loop back mode: 0y0: Normal serial port operation enabled 0y1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

[Explanation]

a. <SOD>

Slave mode output disable. This bit is relevant only in the slave mode (<MS>=1).

14.4.5 SSP0DR (SSP0 data register)

Address = (0x400C_0000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read undefined. Write as zero.
[15:0]	DATA	R/W	0x0000	Transmit/receive FIFO data: 0x00 ~ 0xFF

14.4.6 SSP1DR (SSP1 data register)

Address = (0x400C_1000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:16]	–	–	Undefined	Read undefined. Write as zero.
[15:0]	DATA	R/W	0x0000	Transmit/receive FIFO data: 0x00 ~ 0xFF

[Explanation]

a. <DATA>

Read: Receive FIFO

Write: Transmit FIFO

You must right-justify data when the SSP is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.

14.4.7 SSP0SR (SSP0 status register)

Address = (0x400C_0000) + 0x000C

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read undefined. Write as zero.
[4]	BSY	R	0y0	Busy flag 0y0: Idle 0y1: Busy
[3]	RFF	R	0y0	Receive FIFO full: 0y0: Receive FIFO is not full 0y1: Receive FIFO is full
[2]	RNE	R	0y0	Receive FIFO empty flag 0y0: Receive FIFO is empty 0y1: Receive FIFO is not empty
[1]	TNF	R	0y1	Transmit FIFO full flag: 0y0: Transmit FIFO is full 0y1: Transmit FIFO is not full
[0]	TFE	R	0y1	Transmit FIFO empty flag: 0y0: Transmit FIFO is not empty 0y1: Transmit FIFO is empty

14.4.8 SSP1SR (SSP1 status register)

Address = (0x400C_1000) + 0x000C

Bit	Bit Symbol	Type	Reset Value	Description
[31:5]	–	–	Undefined	Read undefined. Write as zero.
[4]	BSY	R	0y0	Busy flag 0y0: Idle 0y1: Busy
[3]	RFF	R	0y0	Receive FIFO full: 0y0: Receive FIFO is not full 0y1: Receive FIFO is full
[2]	RNE	R	0y0	Receive FIFO empty flag 0y0: Receive FIFO is empty 0y1: Receive FIFO is not empty
[1]	TNF	R	0y1	Transmit FIFO full flag: 0y0: Transmit FIFO is full 0y1: Transmit FIFO is not full
[0]	TFE	R	0y1	Transmit FIFO empty flag: 0y0: Transmit FIFO is not empty 0y1: Transmit FIFO is empty

[Explanation]

a. <BSY>

BSY="1" indicates that the SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.

14.4.9 SSP0CPSR (SSP0 clock prescale register)

Address = (0x400C_0000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read undefined. Write as zero.
[7:0]	CPSDVSR	R/W	0x00	Clock prescale divider: Set an even number from 2 to 254.

14.4.10 SSP1CPSR (SSP1 clock prescale register)

Address = (0x400C_1000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:8]	–	–	Undefined	Read undefined. Write as zero.
[7:0]	CPSDVSR	R/W	0x00	Clock prescale divider: Set an even number from 2 to 254.

[Explanation]

a. <CPSDVSR>

Clock prescale divider. Must be an even number from 2 to 254, depending on the frequency of fsys. The least significant bit always returns zero on reads.

14.4.11 SSP0IMSC (SSP0 interrupt enable/disable register)

Address = (0x400C_0000) + 0x0014

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXIM	R/W	0y0	Transmit FIFO interrupt enable: 0y0: Disable 0y1: Enable
[2]	RXIM	R/W	0y0	Receive FIFO interrupt enable: 0y0: Disable 0y1: Enable
[1]	RTIM	R/W	0y0	Receive timeout interrupt enable: 0y0: Disable 0y1: Enable
[0]	RORIM	R/W	0y0	Receive overrun interrupt enable: 0y0: Disable 0y1: Enable

14.4.12 SSP1IMSC (SSP1 interrupt enable/disable register)

Address = (0x400C_1000) + 0x0014

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXIM	R/W	0y0	Transmit FIFO interrupt enable: 0y0: Disable 0y1: Enable
[2]	RXIM	R/W	0y0	Receive FIFO interrupt enable: 0y0: Disable 0y1: Enable
[1]	RTIM	R/W	0y0	Receive timeout interrupt enable: 0y0: Disable 0y1: Enable
[0]	RORIM	R/W	0y0	Receive overrun interrupt enable: 0y0: Disable 0y1: Enable

[Explanation]

a. <TXIM>
   Enables/disables transmit interrupt
b. <RXIM>
   Enables/disables receive interrupt
c. <RTIM>
   Enables/disables timeout interrupt
d. <RORIM>
   Enables/disables receive overrun interrupt

14.4.13 SSP0RIS (SSP0 pre-enable interrupt status register)

Address = (0x400C_0000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXRIS	R	0y1	Pre-enable transmit interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[2]	RXRIS	R	0y0	Pre-enable receive interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[1]	RTRIS	R	0y0	Pre-enable receive timeout interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[0]	RORRIS	R	0y0	Pre-enable receive overrun interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present

14.4.14 SSP1RIS (SSP1 pre-enable interrupt status register)

Address = (0x400C_1000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXRIS	R	0y1	Pre-enable transmit interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[2]	RXRIS	R	0y0	Pre-enable receive interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[1]	RTRIS	R	0y0	Pre-enable receive timeout interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[0]	RORRIS	R	0y0	Pre-enable receive overrun interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present

14.4.15 SSP0MIS (SSP0 post-enable interrupt status register)

Address = (0x400C_0000) + 0x001C

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXMIS	R	0y0	Post-enable transmit interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[2]	RXMIS	R	0y0	Post-enable receive interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[1]	RTMIS	R	0y0	Post-enable receive timeout interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[0]	RORMIS	R	0y0	Post-enable receive overrun interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present

14.4.16 SSP1MIS (SSP1 post-enable interrupt status register)

Address = (0x400C_1000) + 0x001C

Bit	Bit Symbol	Type	Reset Value	Description
[31:4]	–	–	Undefined	Read undefined. Write as zero.
[3]	TXMIS	R	0y0	Post-enable transmit interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[2]	RXMIS	R	0y0	Post-enable receive interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[1]	RTMIS	R	0y0	Post-enable receive timeout interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present
[0]	RORMIS	R	0y0	Post-enable receive overrun interrupt flag: 0y0: Interrupt not present 0y1: Interrupt present

14.4.17 SSP0ICR (SSP0 interrupt clear register)

Address = (0x400C_0000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RTIC	W	Undefined	Clear the receive timeout interrupt flag: 0y0: Do nothing 0y1: Clear
[0]	RORIC	W	Undefined	Clear the receive overrun interrupt flag: 0y0: Do nothing 0y1: Clear

14.4.18 SSP1ICR (SSP1 interrupt clear register)

Address = (0x400C_1000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RTIC	W	Undefined	Clear the receive timeout interrupt flag: 0y0: Do nothing 0y1: Clear
[0]	RORIC	W	Undefined	Clear the receive overrun interrupt flag: 0y0: Do nothing 0y1: Clear

14.4.19 SSP0DMACR (SSP0DMA control register)

Address = (0x400C_0000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	TXDMAE	R/W	0y0	Transmit FIFO DMA control: 0y0: Disable 0y1: Enable
[0]	RXDMAE	R/W	0y0	Receive FIFO DMA control: 0y0: Disable 0y1: Enable

14.4.20 SSP1DMACR (SSP1DMA control register)

Address = (0x400C_1000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	TXDMAE	R/W	0y0	Transmit FIFO DMA control: 0y0: Disable 0y1: Enable
[0]	RXDMAE	R/W	0y0	Receive FIFO DMA control: 0y0: Disable 0y1: Enable

■ Notes related to specifications

(1) When correct data reception is disturbed due to clock phase shift during reception: After disabling SSP, clearing all data in the receive FIFO and then enabling SSP again will restore the correct reception status.

Example: How to restore a receive data error

```
(SSP0CR1) ←          ((SSP0CR1)&(0xFFFFFFFD))      ; Set "0" to SSP0CR1<SSE>.
                                                   Sync serial port disable
(GPIOAFR1) ←         ((GPIOAFR1)&(0xFFFFFFF0))     ; Set "0" to GPIOAFR1.
                                                   Port A SSP0 function disable
while((SSP0SR)&0x00000004)!=0x00000000){           ; Read (RNE="0")SSP0DR until the receive FIFO
Reg ← (SSP0DR)}                                    becomes empty.
(GPIOAFR1) ←         ((GPIOAFR1)|(0x0000000F))     ; Set "1" to GPIOAFR1.
                                                    Port A SSP0 function enable
(SSP0CR1) ←          ((SSP0CR1)|(0x00000002))      ; Set "1" to SSP0CR1<SSE>.
                                                   Sync serial port enable
```

15 Serial Bus Interface (I2C/SIO)

Important
TMPM382(64 pin version product) do not have SBI1.
Please do not use these function if you use this product.

The TMPM380 contains two Serial Bus Interface (SBI) channels and TMPM382 contains one Serial Bus Interface (SBI) channel, in which the following two operating modes are included:

- I^2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I^2C bus mode, the SBI is connected to external devices via SCL and SDA. In the clock-synchronous 8-bit SIO mode, the SBI is connected to external devices via SCK, SI and SO.

The following table shows the programming required to put the SBI in each operating mode.

As for the Channel 1, it is available for only TMPM380.

Channel	Operation mode	Pin name	Function Control Register	Output Control Register	Input Enable Control Register	Open Drain Control Register
SBI0	I^2C bus mode	SCL0: PC1 SDA0: PC0	PCFR3<1:0> = 11	PCCR<1:0> = 11	PCIE<1:0> = 11	PCOD<1:0> = 11
	SIO mode	SCK0: PC2 SI0 : PC1 SO0 : PC0	PCFR3<2:0> = 111	PCCR<2:0> = 101(SCK0output) PCCR<2:0> = 001(SCK0 input)	PCIE<2:0> = 110	PCOD<2:0> = xxx
SBI1 (only for TMPM380)	I^2C bus mode	SCL1: PG1 SDA1: PG0	PGFR3<1:0> = 11	PGCR<1:0> = 11	PGIE<1:0> = 11	PGOD<1:0> = 11
	SIO mode	SCK1: PG2 SI1 : PG1 SO1 : PG0	PGFR3<2:0> = 111	PGCR<2:0> = 101(SCK1 output) PGCR<2:0> = 001(SCK1 input)	PGIE<2:0> = 110	PGOD<2:0> = xxx

x: Don't care

15.1 Configuration

The configuration is shown in Fig 15-1.

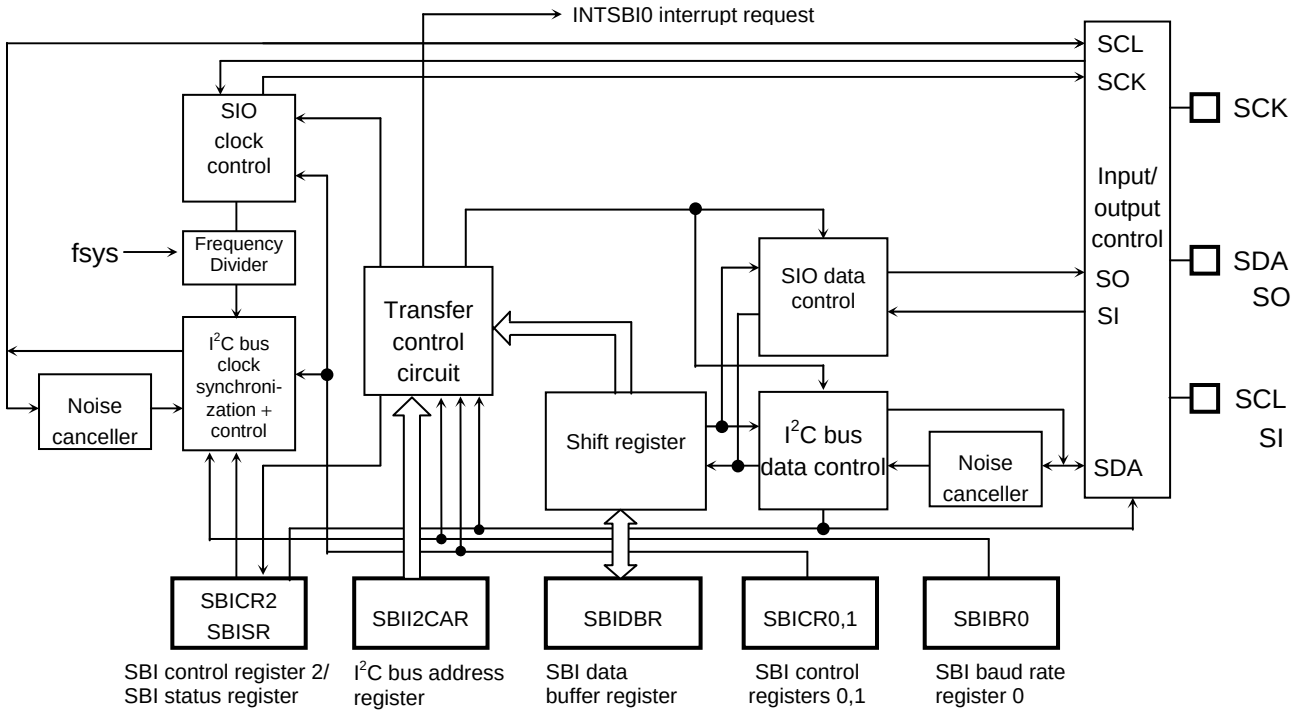


Fig 15-1 SBI Block Diagram

15.2 Control

The following registers control the serial bus interface and provide its status information for monitoring.

- Serial bus interface control registers 0 (SBInCR0)
- Serial bus interface control registers 1 (SBInCR1)
- Serial bus interface control registers 2 (SBInCR2)
- Serial bus interface buffer registers (SBInDBR)
- I^2C bus address register (SBInI2CAR)
- Serial bus interface status registers (SBInSR)
- Serial bus interface baud rate registers 0 (SBInBR0)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to "15.5 Control in the I2C Bus Mode" and "15.7Serial Bus Interface (I2C/SIO)".

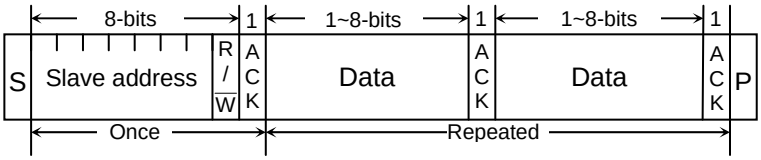
The addresses of each register are shown below.

		Channel 0		Channel 1	
Register name (address)	Serial bus interface control register 0	SBI0CR0	0x4002_0000	SBI1CR0	0x4002_0020
	Serial bus interface control register 1	SBI0CR1	0x4002_0004	SBI1CR1	0x4002_0024
	Serial bus interface control register 2	SBI0CR2 (writing)	0x4002_0010	SBI1CR2 (writing)	0x4002_0030
	Serial bus interface status register	SBI0SR (reading)		SBI1SR (reading)	
	Serial bus interface baud rate register 0	SBI0BR0	0x4002_0014	SBI1BR0	0x4002_0034
	Serial bus interface data buffer register	SBI0DBR	0x4002_0008	SBI1DBR	0x4002_0028
	I^2C bus address register	SBI0I2CAR	0x4002_000C	SBI1I2CAR	0x4002_002C

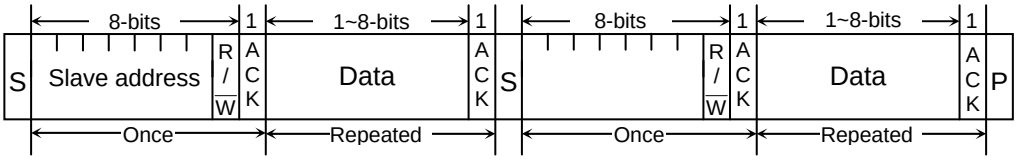
15.3 I2C Bus Mode Data Formats

Fig 15-2 shows the data formats used in the I^2C bus mode.

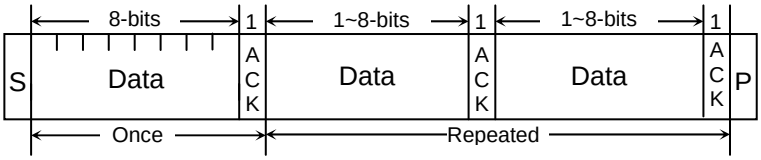
(a) Addressing format



(b) Addressing format (with repeated start condition)



(c) Free data format (master-transmitter to slave-receiver)



Slave address

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

Fig 15-2 I^2C Bus Mode Data Formats

15.4 Control Registers in the I2C Bus Mode

The following registers control the serial bus interface (SBI) in the I^2C bus mode and provide its status information for monitoring.

Serial bus control register 0

SBInCR0

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	This can be read as “0.”						

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

<SBIEN>: To use the SBI, enable the SBI operation (“1”) before setting each register in the SBI module. For the first time in case of setting to enable, the relevant SBI registers can be read or written.

Fig 15-3 I^2C Bus Mode register

Serial bus control register 1

SBInCR1

	7	6	5	4	3	2	1	0
Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	R/W			R/W	R	R/W		R/W
After reset	0	0	0	0	1	0	0	1
Function	Select the number of bits per transfer (Note 1)			Acknowledgment clock 0: Not generate 1: Generate	This can be read as "1."	Select internal SCL output clock frequency (Note 2) and reset monitor.		

	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

<Bit 2:0><SCK2:0>: Select internal SCL output clock frequency

On writing <SCK2:0>: Select internal SCL output clock frequency

000	n=5	384 kHz
001	n=6	294 kHz
010	n=7	200 kHz
011	n=8	122 kHz
100	n=9	68 kHz
101	n=10	36 kHz
110	n=11	19 kHz
111		reserved

System clock: fsys (=40 MHz)

Clock gear : fc/1

$$\text{Frequency} = \frac{\text{fsys}}{2^n + 72} \text{ [Hz]}$$

<Bit 0>< SWRMON>: Software reset status monitor

On reading <SWRMON>: Software reset status monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

<Bit 7:5><BC2:0>: Select the number of bits per transfer

Select the number of bits per transfer

<BC2:0>	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

Fig 15-4 I^2C Bus Mode register

(Note 1)	**Clear <BC2:0> to "000" before switching the operation mode to the clock-synchronous 8-bit SIO mode.**
(Note 2)	**For details on the SCL line clock frequency, refer to "15.5.3** Serial Bus Interface (I2C/SIO)**."**
(Note 3)	**After a reset, the <SCK0/SWRMON> bit is read as "1." However, if the SIO mode is selected at the SBInCR2 register, the initial value of the <SCK0> bit is "0."**
(Note 4)	**Initial value for frequency select is <SCK[2:0]>=000, and there is no relation to the initial reading value.**

Serial bus control register 2

SBInCR2

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 2)		W (Note 1)	
After reset	0	0	0	1	0	0	0	0
Function	Select master/slave 0: Slave 1: Master	Select transmit/ receive 0: Receive 1: Transmit	Start/stop condition generation 0: Stop condition generated 1: Start condition generated	Clear INTSBIn interrupt request 0: – 1: Clear interrupt request	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I2C bus mode 11: (Reserved))		Software reset generation Write “10” followed by “01” to generate a reset.	

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

<Bit 1:0><SWRST1:0>: Write “10” followed by “01” to generate a reset.
<Bit 3:2><SBIM1:0> : Select serial bus interface operating mode

Select serial bus interface operating mode (Note 2)

00	Port mode (disables serial bus interface output)
01	Clock –synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

<Bit 4><PIN> : Clear INTSBIn interrupt request
<Bit 5><BB> : Start/stop condition generation
<Bit 6><TRX> : Select transmit/ receive
<Bit 7><MST> : Select master/slave

(Note 1) Reading this register causes it to function as the SBInSR register.

(Note 2) Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the “H” level before switching the operating mode from the port mode to the I^2C bus or clock-synchronous 8-bit SIO mode.

(Note 3) Ensure that serial transfer is completed before switching the mode.

Fig 15-5 I^2C Bus Mode register

Table 15-1 Base Clock Resolution

@fsys = 40 MHz

Clock gear value <GEAR1:0>	Base clock resolution
00 (fc)	fsys/2^2 (0.1 us)
01 (fc/2)	fsys/2^3 (0.2 us)
10 (fc/4)	fsys/2^4 (0.4 us)
11 (fc/8)	fsys/2^5 (0.8 us)

Serial bus interface status register

SBInSR

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	AL	AAS	ADO	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/ slave selection monitor 0: Slave 1: Master	Transmit/ receive selection monitor 0: Receive 1: Transmit	I^2C bus state monitor 0: Free 1: Busy	INTSBIn interrupt request monitor 0: Interrupt request generated 1: Interrupt request cleared	Arbitration lost detection 0: – 1: Detected	Slave address match detection 0: – 1: Detected	General call detection 0: – 1: Detected	Last received bit monitor 0: "0" 1: "1"

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

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

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

(Note) Writing to this register causes it to function as SBInCR2.

Fig 15-6 I^2C Bus Mode register

<Bit 0><LRB> : Last received bit monitor

<Bit 1><ADO> : General call detection

<Bit 2><AAS> : Slave address match detection

<Bit 3><AL> : Arbitration lost detection

<Bit 4><PIN> : INTSBIn interrupt request monitor

<Bit 5><BB> : I^2C bus state monitor

<Bit 6><TRX> : Transmit/ receive selection monitor

<Bit 7><MST> : Master/ slave selection monitor

Serial bus interface baud rate register 0

SBInBR0

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI						
Read/Write	R	R/W	R					R/W
After reset	1	0	1					0
Function	This can be read as “1”.	IDLE 0: Stop 1:Operate	This can be read as “1”.					Be sure to write “0.” (Note)

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

<Bit 6><I2SBI> : Operation at the IDLE mode

Serial bus interface data buffer register

SBInDBR

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	0							
Function	RX data/ TX data.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

(Note1) **The transmission data must be written in to the register from the MSB (bit 7).**

The received data is stored in the LSB.

(Note2) **Since SBIxI2CAR 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.**

I²C bus address register

SBInI2CAR

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set the slave address when the SBI acts as a slave device.							Specify address recognition mode

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

<Bit 0><ALS> : Specify address recognition mode

(Note1) **Please set the bit 0 <ALS> of I2C bus address register SBInI2CAR 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.**

(Note2) **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.)**

Fig 15-7 I²C Bus Mode Register

15.5 Control in the I2C Bus Mode

15.5.1 Setting the Acknowledgement Mode

Setting SBInCR1<ACK> to “1” selects the acknowledge mode. When operating as a master, the SBI adds one clock for acknowledgment signals. As a transmitter, the SBI releases the SDA pin during this clock cycle to receive acknowledgment signals from the receiver. As a receiver, the SBI pulls the SDA pin to the “L” level during this clock cycle and generates acknowledgment signals.

By setting <ACK> to “0”, the non-acknowledgment mode is activated. When operating as a master, the SBI does not generate clock for acknowledgement signals.

15.5.2 Setting the Number of Bits per Transfer

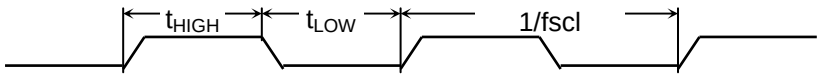
SBInCR1 <BC2:0> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC2:0> is set to “000,” causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC2:0> keeps a previously programmed value.

15.5.3 Serial Clock

① Clock source

SBInCR1 <SCK2:0> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$t_{LOW} = 2^{n-1}/fsys + 58/ fsys$

$t_{HIGH} = 2^{n-1}/ fsys + 14/ fsys$

$fscl = 1/(t_{Low} + t_{HIGH})$

$= \dfrac{fsys}{2^n + 72}$

SBInCR1 <SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Fig 15-8 Clock Source

(Note)	**The highest speeds in the standard and high-speed modes are specified to 100kHz and 400kHz respectively following the communications standards. Note that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.**

② Clock Synchronization

The I^2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the “L” level overrides other masters producing the “H” level on their clock lines. This must be detected and responded by the masters producing the “H” 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.

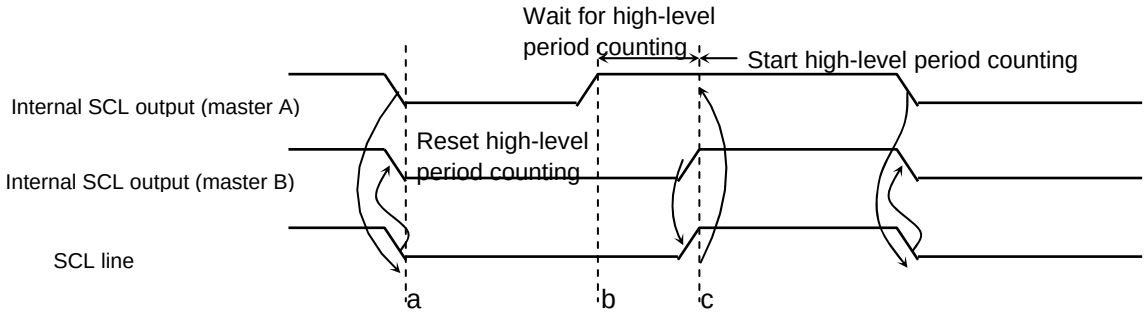


Fig 15-9 Example of Clock Synchronization

At the point a, Master A pulls its internal SCL output to the “L” level, bringing the SCL bus line to the “L” level. Master B detects this transition, resets its “H” level period counter, and pulls its internal SCL output level to the “L” level.

Master A completes counting of its “L” level period at the point b, and brings its internal SCL output to the “H” level. However, Master B still keeps the SCL bus line at the “L” level, and Master A stops counting of its “H” level period counting. After Master A detects that Master B brings its internal SCL output to the “H” level and brings the SCL bus line to the “H” level at the point c, it starts counting of its “H” level period.

This way, the clock on the bus is determined by the master with the shortest “H” level period and the master with the longest “L” level period among those connected to the bus.

15.5.4 Slave Addressing and Address Recognition Mode

When the SBI is configured to operate as a slave device, the slave address <SA6:0> and <ALS> must be set at SBInI2CAR. Setting <ALS> to “0” selects the address recognition mode.

15.5.5 Configuring the SBI as a Master or a Slave

Setting SBInCR2<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.

15.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBInCR2 <TRX> to “1” configures the SBI as a transmitter. Setting <TRX> to “0” configures the SBI as a receiver.

At the slave mode, the SBI receives the direction bit ($R/\overline{W}$) from the master device on the following occasions:

- when data is transmitted in the addressing format
- when the received slave address matches the value specified at I2CCR
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros

If the value of the direction bit ($R/\overline{W}$) is “1,” <TRX> is set to “1” by the hardware. If the bit is “0,” <TRX> is set to “0”.

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of “1” is transmitted, <TRX> is set to “0” by the hardware. If the direction bit is “0,” <TRX> changes to “1.” If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to “0” by the hardware when it detects the stop condition on the bus or the arbitration lost.

15.5.7 Generating Start and Stop Conditions

When SBInSR<BB> is “0,” writing “1” to SBInCR2 <MST, TRX, BB, PIN> causes the SBI to generate the start condition on the bus and output 8-bit data. <ACK> must be set to “1” in advance.

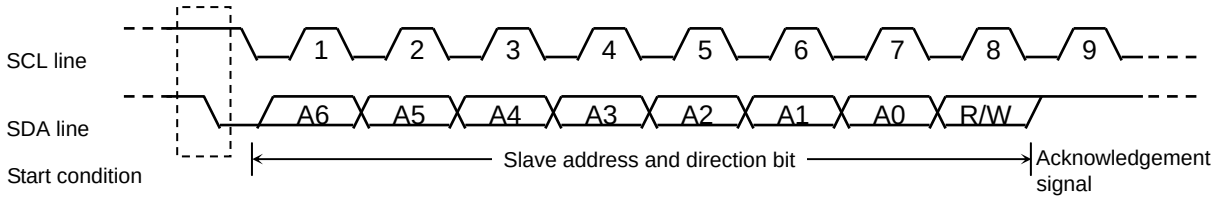


Fig 15-10 Generating the Start Condition and a Slave Address

When <BB> is “1,” writing “1” to <MST, TRX, PIN> and “0” to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

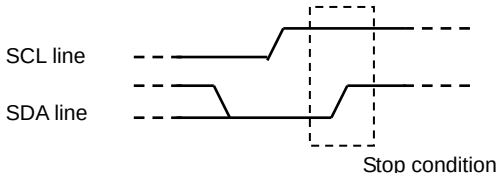


Fig 15-11 Generating the Stop Condition

SBInSR<BB> can be read to check the bus state. <BB> is set to “1” when the start condition is detected on the bus (the bus is busy), and set to “0” when the stop condition is detected (the bus is free).

15.5.8 Interrupt Service Request and Release

When a serial bus interface interrupt request (INTSBIn) is generated, SBInCR2 <PIN> is cleared to “0.” While <PIN> is “0,” the SBI pulls the SCL line to the “L” level.

After transmission or reception of one data word, <PIN> is cleared to “0.” It is set to “1” when data is written to or read from SBInDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to “1.”

In the address recognition mode (<ALS> = “0”), <PIN> is cleared to “0” when the received slave address matches the value specified at SBInI2CAR or when a general-call address is received; i.e., the eight bits following the start condition are all zeros. When the program writes “1” to SBInCR2<PIN>, it is set to “1.” However, writing “0” does clear this bit to “0”.

15.5.9 Serial Bus Interface Operating Modes

SBInCR2 <SBIM1:0> selects an operating mode of the serial bus interface. <SBIM1:0> must be set to “10” to configure the SBI for the I^2C bus mode. Make sure that the bus is free before switching the operating mode to the port mode.

15.5.10 Lost-arbitration Detection Monitor

The I^2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines. The I^2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below. Up until the point a, Master A and Master B output the same data. At the point a, Master A outputs the “L” level and Master B outputs the “H” level. Then Master A pulls the SDA bus line to the “L” 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 “Lost Arbitration”. Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

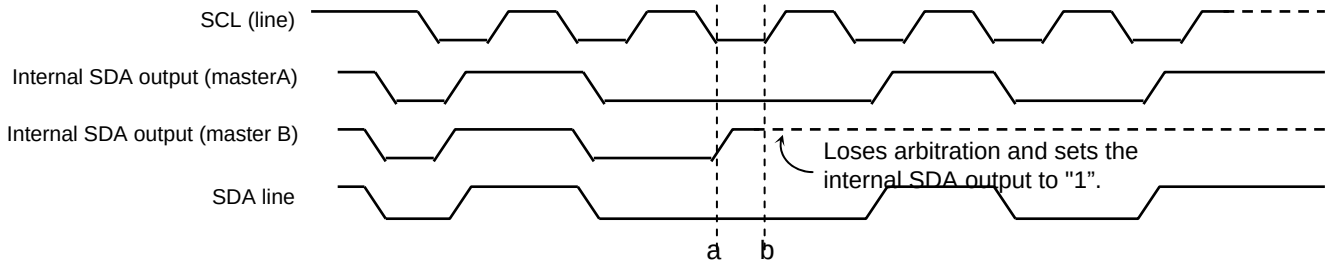


Fig 15-12 Lost Arbitration

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 SBInSR <AL> is set to “1”.

When <AL> is set to “1,” SBInSR <MST, TRX> are cleared to “0,” causing the SBI to operate as a slave receiver. <AL> is cleared to “0” when data is written to or read from SBInDBR or data is written to SBInCR2.

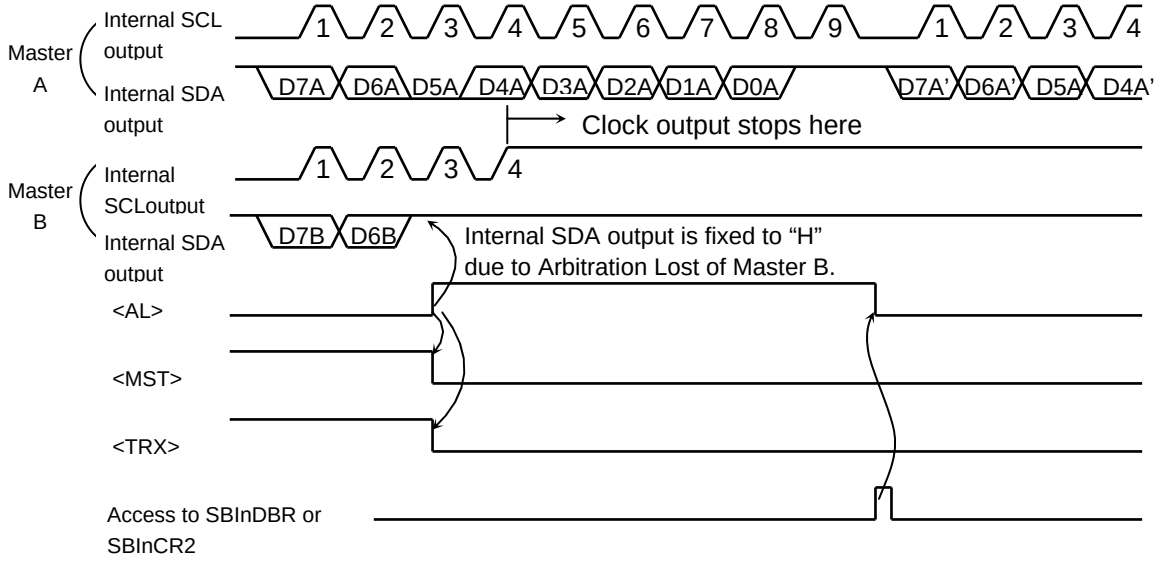


Fig 15-13 Example of Master B Lost Arbitration (D7A = D7B, D6A = D6B)

15.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (SBInI2CAR <ALS> = ”0”), SBInSR <AAS> is set to “1” on receiving the general-call address or the slave address that matches the value specified at SBInI2CAR. 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 SBInDBR.

15.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBInSR <AD0> is set to “1” when it receives the general-call address; i.e., the eight bits following the start condition are all zeros. <AD0> is cleared to “0” when the start or stop condition is detected on the bus.

15.5.13 Last Received Bit Monitor

SBInSR <LRB> is set to the SDA line value that was read at the rising of the SCL line. In the acknowledgment mode, reading SBInSR <LRB> immediately after generation of the INTSBIn interrupt request causes ACK signal to be read.

15.5.14 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing “10” followed by “01” to SBInCR2 <SWRST1:0> generates a reset signal that initializes the serial bus interface circuit. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to “0”.

(Note) A software reset causes the SBI operating mode to switch from the I^2C mode to the port mode.

15.5.15 Serial Bus Interface Data Buffer Register (SBInDBR)

Reading or writing SBInDBR initiates reading received data or writing transmitted data. When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

15.5.16 I2C Bus Address Register (SBInI2CAR)

When the SBI is configured as a slave device, the SBInI2CAR<SA6:0> bit is used to specify a slave address. If SBInI2CAR <ALS> is set to “0,” 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.

15.5.17 Baud Rate Register (SBInBR0)

The SBInBR0<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.

15.6 Data Transfer Procedure in the I2C Bus Mode

15.6.1 Device Initialization

First, program SBInCR1<ACK, SCK2:0> by writing “0” to bits 7 to 5 in SBInCR1.

Next, program SBInI2CAR by specifying a slave address at <SA6:0> and an address recognition mode at <ALS>. (<ALS> must be set to”0” when using the addressing format).

Then program SBInCR2 to initially configure the SBI in the slave receiver mode by writing “0” to <MST, TRX, BB> , “1” to <PIN> , “10” to <SBIM1:0> and “0” to bits 1 and 0.

		7	6	5	4	3	2	1	0	
SBInCR1	←	0	0	0	X	0	X	X	X	Specifies ACK and SCL clock.
SBInI2CAR	←	X	X	X	X	X	X	X	X	Specifies a slave address and an address recognition mode.
SBInCR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

(Note) X: Don’t care

15.6.2 Generating the Start Condition and a Slave Address

① Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = “0”). Then, write “1” to SBInCR1 <ACK> to select the acknowledgment mode. Write to SBInDBR a slave address and a direction bit to be transmitted.

When <BB> = “0,” writing “1111” to SBInCR2 <MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBInDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBIn interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to ”0.” In the master mode, the SBI holds the SCL line at the “L” level while <PIN> is “0.” <TRX> changes its value according to the transmitted direction bit at generation of the INTSBIn interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Settings in main routine

```
                    7 6 5 4 3 2 1 0
┌→ Reg.       ← SBISR
│  Reg.       ← Reg. e 0x20
│  if Reg.    ≠ 0x00                    Ensures that the bus is free.
└─ Then
   SBInCR1  ← X X X 1 0 X X X           Selects the acknowledgement mode.
   SBInDBR  ← X X X X X X X X           Specifies the desired slave address and direction.
   SBInCR2  ← 1 1 1 1 1 0 0 0           Generates the start condition.
```

Example of INTSBIn interrupt routine

```
Clears the interrupt request.
Processing
End of interrupt
```

② Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line. If the received address matches its slave address specified at SBInI2CAR or is equal to the general-call address, the SBI pulls the SDA line to the “L” level during the ninth clock and outputs an acknowledgment signal.

The INTSBIn interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to “0.” In the slave mode, the SBI holds the SCL line at the “L” level while <PIN> is “0”.

(Note) The user can only use a DMA transfer:
- **when there is only one master and only one slave and**
- **continuous transmission or reception is possible.**

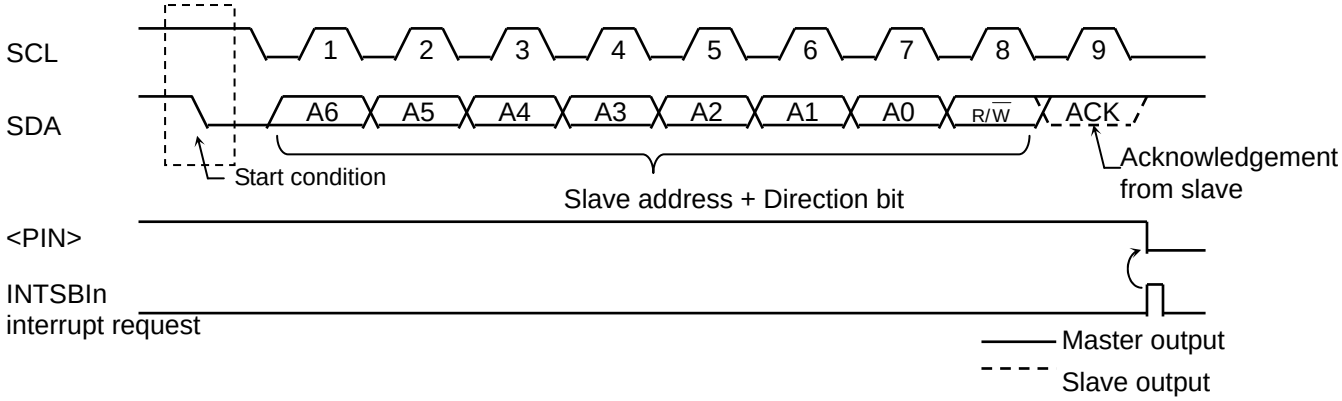


Fig 15-14 Generation of the Start Condition and a Slave Address

15.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBIn 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.

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 SBInDBR. If the data has different length, <BC2:0> and <ACK> are programmed and the transmit data is written into SBInDBR. Writing the data makes <PIN> to”1,” causing the SCL pin to generate a serial clock for transferring a next data word, and the SDA pin to transfer the data word. After the transfer is completed, the INTSBIn interrupt request is generated, <PIN> is set to “0,” and the SCL pin is pulled to the “L” level. To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBIn 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

SBInCR1 ← X X X X 0 X X X	Specifies the number of bits to be transmitted and specify whether ACK is required.
SBInDBR ← X X X X X X X X	Writes the transmit data.

End of interrupt processing

(Note) X: Don’t care

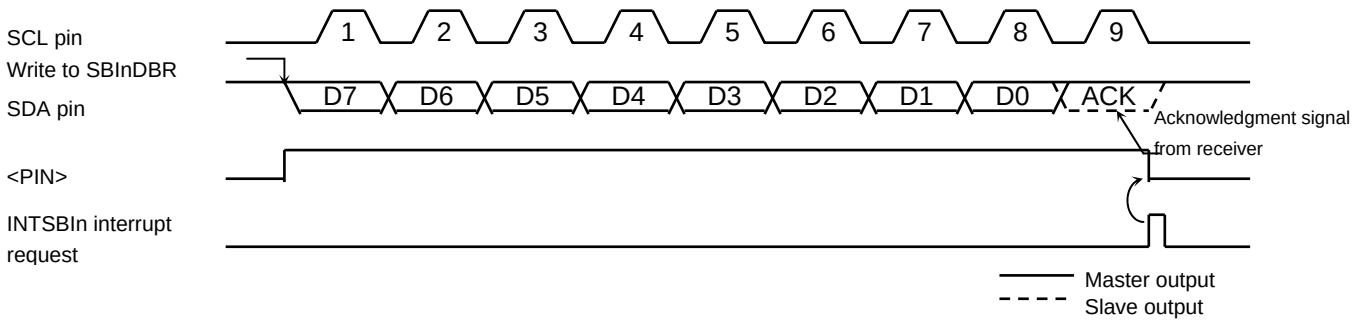


Fig 15-15 <BC2:0> = “000” and <ACK> = “1” (Transmitter Mode)

Receiver mode (<TRX> = “0”)

If the next data to be transmitted has eight bits, the transmit data is written into SBInDBR. If the data has different length, <BC2:0> and <ACK> are programmed and the received data is read from SBInDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.) On reading the data, <PIN> is set to “1,” and the serial clock is output to the SCL pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the “L” level, “0” is output to the SDA pin.

After that, the INTSBIn interrupt request is generated, and <PIN> is cleared to “0,” pulling the SCL pin to the “L” level. Each time the received data is read from SBInDBR, one-word transfer clock and an acknowledgement signal are output.

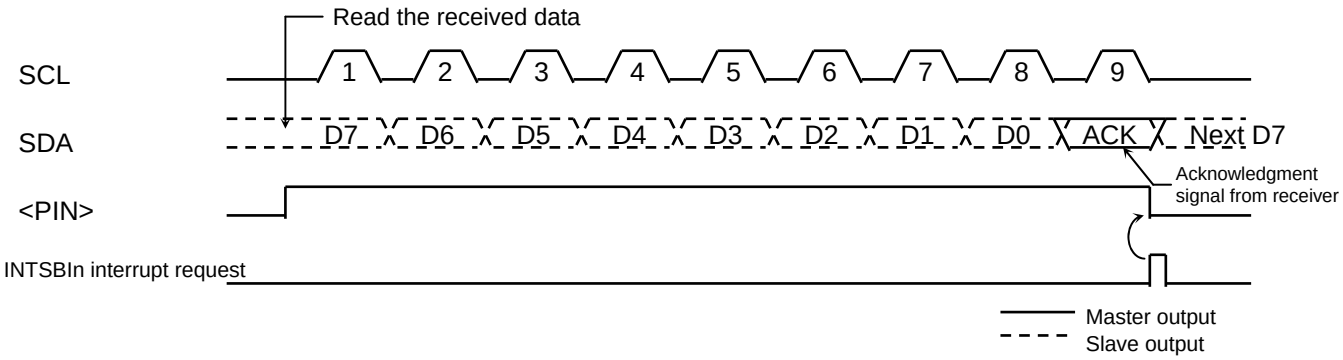


Fig 15-16 <BC2:0> = “000” and <ACK> = “1” (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be set 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, <BC2:0> 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 “H” 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.

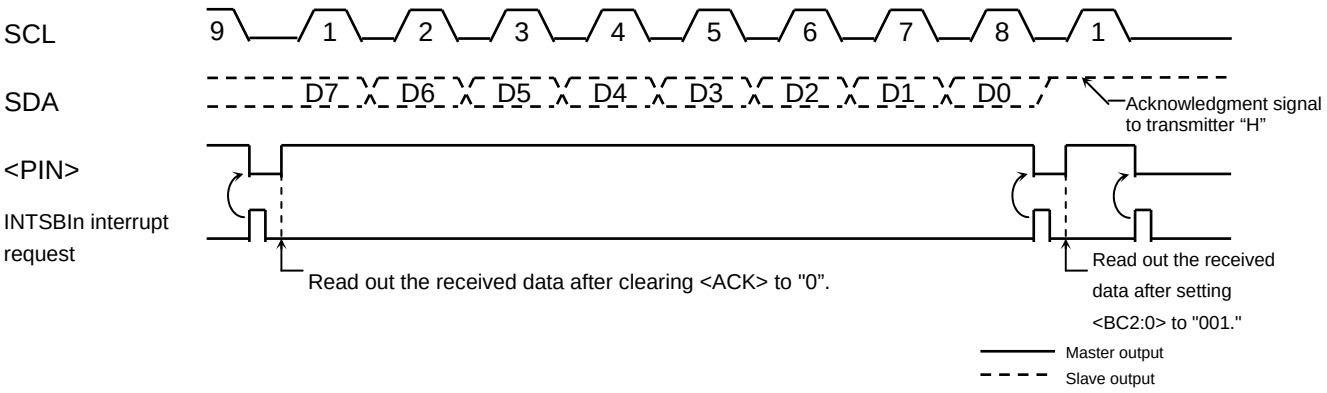


Fig 15-17 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data word

INTSBIn interrupt (after data transmission)

```
                  7 6 5 4 3 2 1 0
SBInCR1  ← 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.     ← SBInDBR              Reads dummy data.
End of interrupt
```

INTSBIn interrupt (first to (N-2)th data reception)

```
                  7 6 5 4 3 2 1 0
Reg.     ← SBInDBR              Reads the first to (N-2)th data words.
End of interrupt
```

INTSBIn interrupt ((N-1)th data reception)

```
                  7 6 5 4 3 2 1 0
SBInCR1  ← X X X 0 0 X X X      Disables generation of acknowledgement clock.
Reg.     ← SBInDBR              Reads the (N-1)th data word.
End of interrupt
```

INTSBIn interrupt (Nth data reception)

```
                  7 6 5 4 3 2 1 0
SBInCR1  ← 0 0 1 0 0 X X X      Disables generation of acknowledgement clock.
Reg.     ← SBInDBR              Reads the Nth data word.
End of interrupt
```

INTSBIn 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, the SBI generates the INTSBIn interrupt request on four occasions: 1) when the SBI has received any slave address from the master, 2) when the SBI has received a general-call address, 3) when the received slave address matches its own address, and 4) when a data transfer has been completed in response to a general-call. Also, if the SBI detects Arbitration Lost in the master mode, it switches to the slave mode. Upon the completion of data word transfer in which Arbitration Lost is detected, the INTSBIn interrupt request is generated, <PIN> is cleared to “0,” and the SCL pin is pulled to the “L” level. When data is written to or read from SBInDBR or when <PIN> is set to “1,” the SCL pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of Arbitration Lost is carried out.

SBInSR <AL>, <TRX>, <AAS> and <AD0> are tested to determine the processing required.

Table 15-2 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.

INTSBIn interrupt

```
if TRX = 0
Then go to other processing
if AL = 1
Then go to other processing
if AAS = 0
Then go to other processing
SBInCR1  ← X X X 1 0 X X X      Sets the number of bits to be transmitted.
SBInDBR  ← X X X X 0 X X X      Sets the transmit data.
```

(Note) X: Don’t care

Table 15-2 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 <BC2:0> and write the transmit data into SBInDBR.
	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 <BC2:0> and write the transmit data to the SBInDBR.
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 SBInDBR (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 <BC2:0> and read the received data from SBInDBR.

15.6.4 Generating the Stop Condition

When SBInSR <BB> is “1,” writing “1” to SBInCR2 <MST, TRX, PIN> and “0” to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released. After that, the SDA pin goes high, causing the stop condition to be generated.

	7	6	5	4	3	2	1	0	
SBInCR2 ←	1	1	0	1	1	0	0	0	Generates the stop condition.

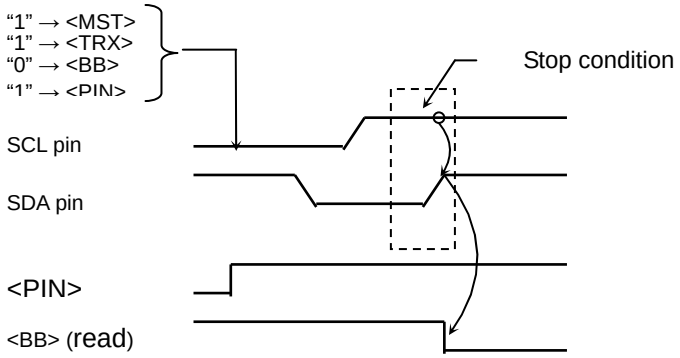


Fig 15-18 Generating the Stop Condition

15.6.5 Restart Procedure

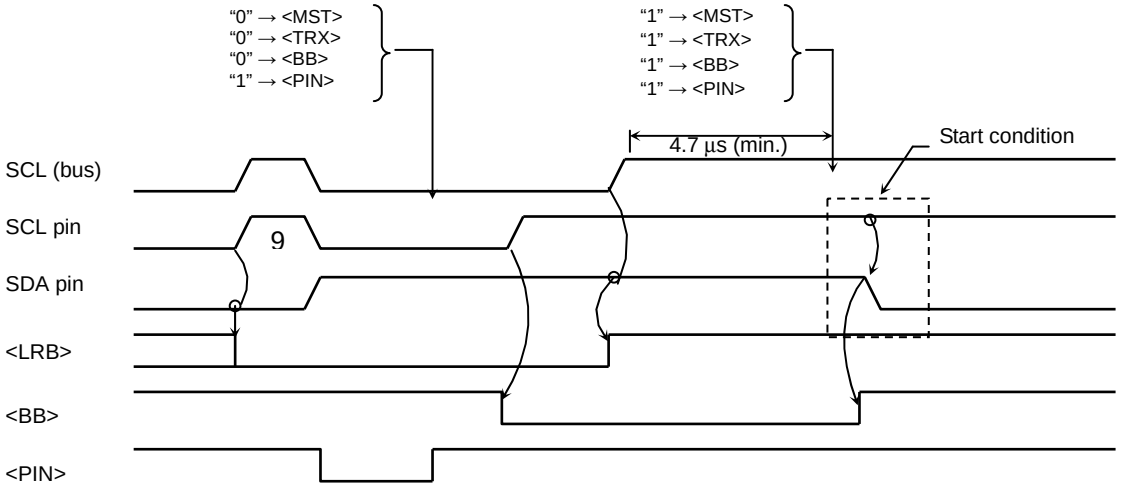
Restart is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a restart in the master mode is described below.

First, set SBInCR2 <MST, TRX, BB> to “0” and write “1” to <PIN> to release the bus. At this time, the SDA pin is held at the “H” level and the SCL pin is released. Because no stop condition is generated on the bus, other devices recognize that the bus is busy. Then, test SBInSR <BB> and wait until it becomes “0” to ensure that the SCL 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 “L” level. Once the bus is determined to be free this way, use the above-mentioned steps 15.6.2 to generate the start condition.

To satisfy the setup time of restart, at least 4.7-μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

	7 6 5 4 3 2 1 0	
SBInCR2	← 0 0 0 1 1 0 0 0	Releases the bus.
if SBInSR<BB> ≠ 0		Checks that the SCL pin is released.
Then		
if SBInSR<LRB> ≠1		Checks that no other device is pulling the SCL pin to the “L” level.
Then		
4.7 μs Wait		
SBInCR1	← X X X 1 0 X X X	Selects the acknowledgment mode.
SBInDBR	← X X X X X X X X	Sets the desired slave address and direction.
SBInCR2	← 1 1 1 1 1 0 0 0	Generates the start condition.

(Note) X: Don’t care



(Note) Do not write <MST> to “0” when it is “0.” (Restart cannot be initiated.)

Fig 15-19 Timing Chart of Generating a Restart

15.7 Control in the Clock-synchronous 8-bit 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.

Serial bus interface control register 0

SBInCR0

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	This can be read as “0.”						

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

<SBIEN>: To use the SBI, enable the SBI operation (“1”) before setting each register of SBI module.

Fig 15-20 SIO Mode Registers

Serial bus interface control register 1

SBInCR1

	7	6	5	4	3	2	1	0
bit Symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	R/W				R	R/W		
After reset	0	0	0	0	1	0	0	0
Function	Start transfer 0: Stop 1: Start	Transfer 0: Continue 1: Forced terminatio n	Select transfer mode 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode		This can be read as “1”.	Select serial clock frequency		

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

On writing <SCK2:0>: Select serial clock frequency

000	n = 3	2.5 MHz
001	n = 4	1.25 MHz
010	n = 5	625 kHz
011	n = 6	313 kHz
100	n = 7	156 kHz
101	n = 8	78 kHz
110	n = 9	39 kHz
111	—	External clock

System clock : fsys (=40 MHz)

Clock gear : fc/1

Frequency $= \frac{fsys/2}{2^n}$ [Hz]

(Note) Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.

Fig 15-21 SIO Mode Registers

Serial bus interface data buffer register

SBInDBR

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	Undefined							
Function	RX data/ TX data							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
リセット後	0							
機　能	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

Serial bus interface control register 2

SBInCR2

	7	6	5	4	3	2	1	0
bit Symbol					SBIM1	SBIM0		
Read/Write	R				W		R	
After reset	1				0	0	1	
Function	This can be read as “1”.				Select serial bus interface operating mode 00: Port mode 01: Clock-synchronous 8-bit SIO mode 10: I^2C bus mode 11: (Reserved)		This can be read as “1”.	
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

Fig 15-22 SIO Mode Registers

Serial bus interface register

SBInSR

	7	6	5	4	3	2	1	0
bit Symbol					SIOF	SEF		
Read/Write	R				R		R	
After reset	1				0	0	1	
Function	This can be read as “1”.				Serial transfer status monitor 0: Completed 1: In progress	Shift operation status monitor 0: Completed 1: In progress	This can be read as “1”.	

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

Serial bus interface baud rate register 0

SBInBR0

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI						
Read/Write	R	R/W	R					R/W
After reset	1	0	1					0
Function	This can be read as “1”.	IDLE 0: Stop 1: Operate	This can be read as “1”.					Make sure to write “0.”

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”							

Fig 15-23 SIO Mode Registers

15.7.1 Serial Clock

① Clock source

Internal or external clocks can be selected by programming SBInCR1 <SCK2:0>.

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 SCK pin. At the beginning of a transfer, the SCK pin output becomes the “H” 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.

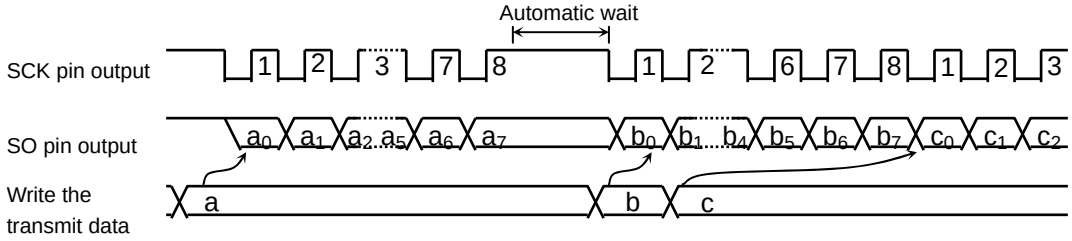
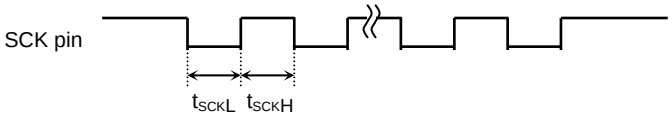


Fig 15-24 Automatic Wait

External clock (<SCK2:0> = “111”)

The SBI uses an external clock supplied from the outside to the SCK pin as a serial clock. For proper shift operations, the serial clock at the “H” and “L” levels must have the pulse widths as shown below.



t_{SCKL}, t_{SCKH} > 4/fsys

Fig 15-25 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 SCK pin input/output).

Trailing-edge shift

Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCK pin input/output).

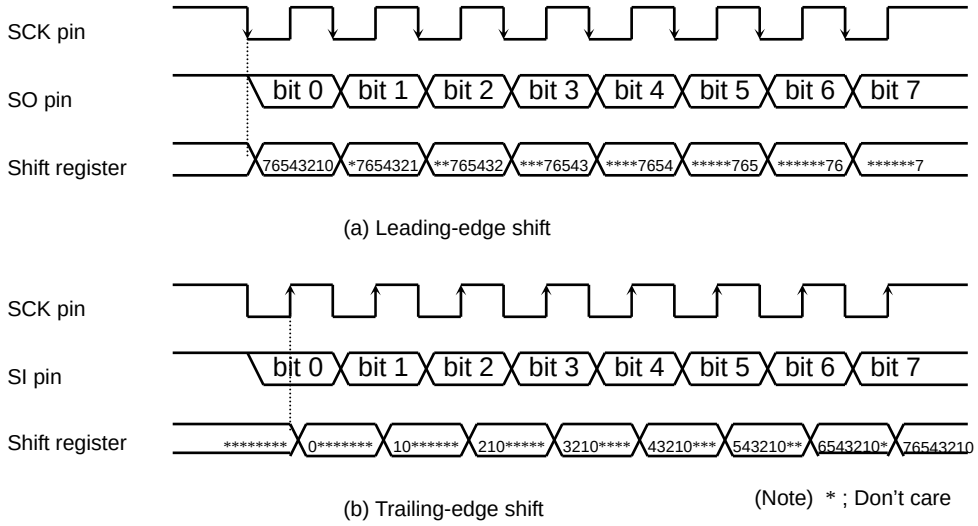


Fig 15-26 Shift Edge

15.7.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBInCR1 <SIOM1:0>.

① 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBInDBR.

After writing the transmit data, writing “1” to SBInCR1 <SIOS> starts the transmission. The transmit data is moved from SBInDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBInDBR becomes empty, and the INTSBIn (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 SBInDBR is loaded with the next transmit data.

In the external clock mode, SBInDBR 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 SBInDBR 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 SBInSR <SIOF> to “1” to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to “0” or setting <SIOINH> to “1” in the INTSBIn interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBInSR <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”.

In the external clock mode, <SIOS> must be set to “0” before the next transmit data shift operation is started. Otherwise, operation will stop after dummy data is transmitted.

```
                7 6 5 4 3 2 1 0
SBInCR1   ← 0 1 0 0 0 X X X      Selects the transmit mode.

SBInDBR   ← X X X X X X X X      Writes the transmit data.
SBInCR1   ← 1 0 0 0 0 X X X      Starts transmission.

INTSBIn interrupt
SBInDBR   ← X X X X X X X X      Writes the transmit data.
```

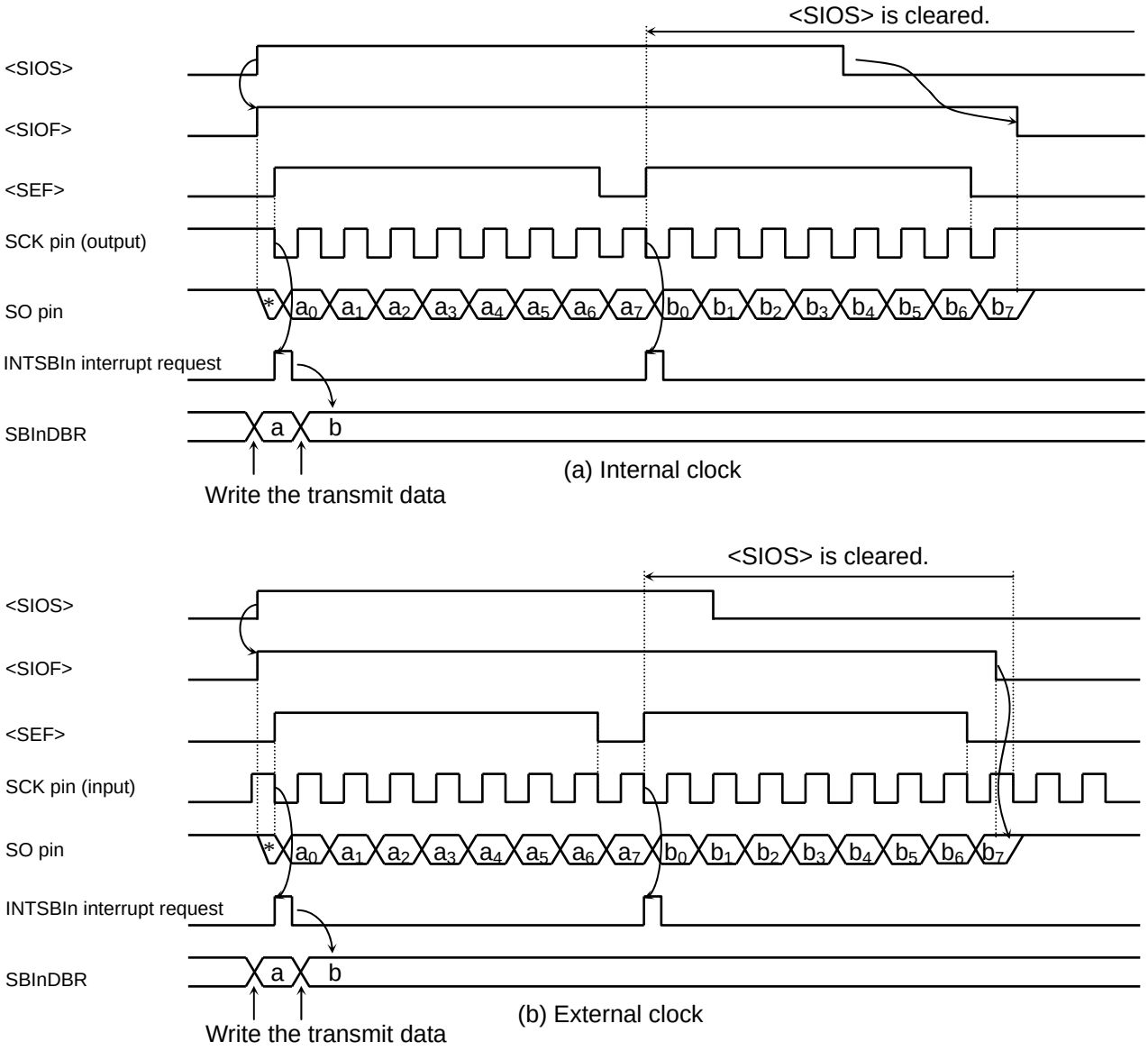


Fig 15-27 Transmit Mode

Example: Example of programming (external clock) to terminate transmission by <SIO>

```
                    7 6 5 4 3 2 1 0
if SBISR<SIOF> ≠ 0                    Recognizes the completion of the transmission.
Then
if SCK ≠ 1                            Recognizes “1” is set to the SCK pin by monitoring the port.
Then
SBInCR1 ← 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 SBInCR1 <SIOS> enables reception. Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBInDBR and the INTSBIn (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBInDBR.

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 SBInDBR.

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 INTSBIn interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBInDBR. The program checks SBInSR <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 SBInDBR 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
SBInCR1  ← 0 1 1 1 0 X X X        Selects the receive mode.

SBInCR1  ← 1 0 1 1 0 0 0 0        Starts reception.
```

INTSBIn interrupt

```
Reg.     ← SBInDBR                Reads the received data.
```

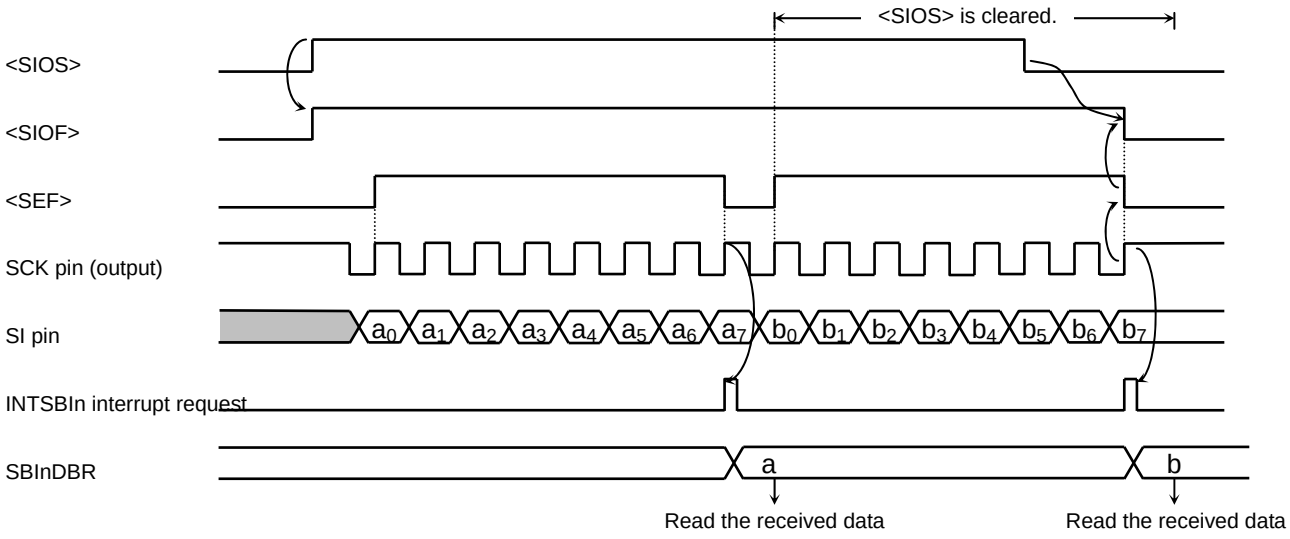


Fig 15-28 Receive Mode (Example: Internal Clock)

③ 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBInDBR and setting SBInCR1 <SIOS> to “1” enables transmission and reception. The transmit data is output through the SO pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBInDBR and the INTSBIn interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBInDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started. The maximum data transfer rate for the external clock operation varies depending on the maximum latency between when the interrupt request is generated and when the transmit data is written.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to “1” to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to “0” or setting SBInCR1 <SIOINH> to “1” in the INTSBIn interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBInDBR. The program checks SBInSR <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, the transmission and reception are aborted immediately and <SIOF> is cleared to “0.”

(Note) The contents of SBInDBR 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.

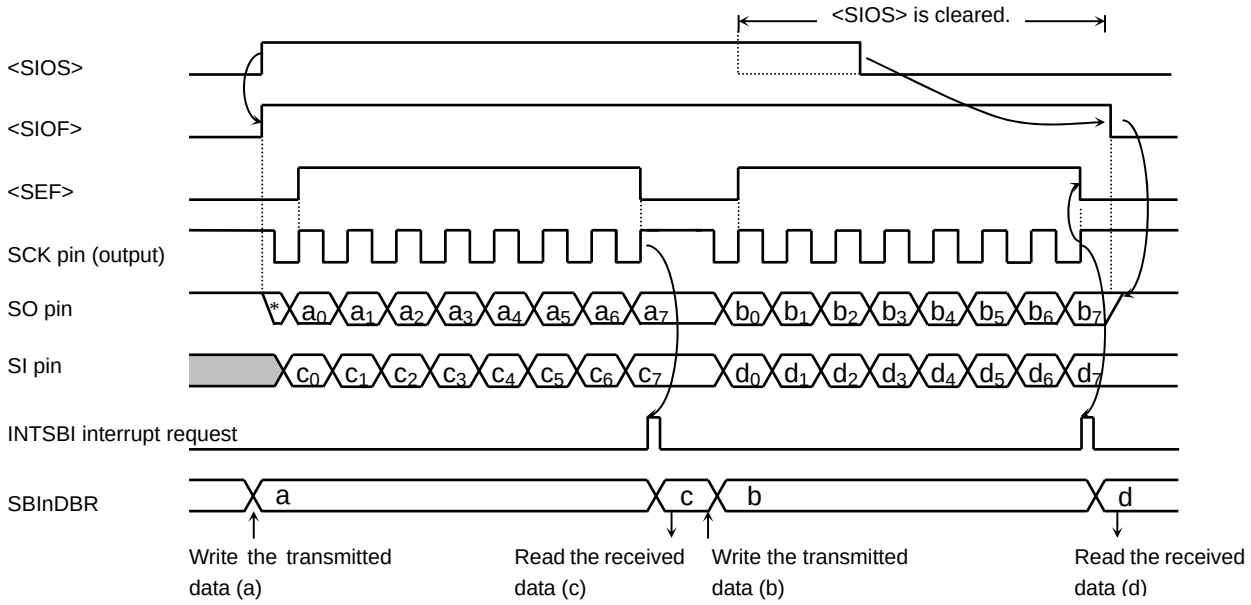


Fig 15-29 Transmit/Receive Mode (Example: Internal Clock)

```
                  7 6 5 4 3 2 1 0
SBInCR1  ← 0 1 1 0 0 X X X        Selects the transmit mode.

SBInDBR  ← X X X X X X X X        Writes the transmit data.
SBInCR1  ← 1 0 1 0 0 X X X        Starts reception/transmission.
```

INTSBIn interrupt

```
Reg.     ← SBInDBR                Reads the received data.
SBInDBR  ← 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 SCK rising edge as shown below. Transmit mode and transmit/receive mode are the same.

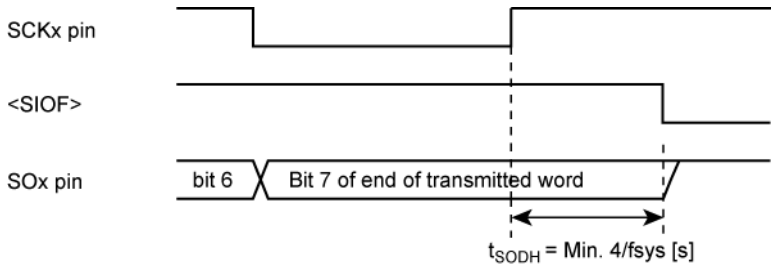


Fig 15-30 Data retention time of the last bit at the end of transmission

16 Remote control signal preprocessor (RMC)

16.1 Basic operation

Remote control signal preprocessor (hereafter referred to as RMC) receives a remote control signal of which carrier is removed.

16.1.1 Reception of Remote Control Signal

- Sampled by 32kHz clock
- Noise canceller
- Leader detection
- Batch reception up to 72bit of data

16.2 Registers

16.2.1 Register Map

Addresses and names of RMC control registers are shown below.

Register		Address
Remote Control Enable Register	RMCEN	0x4004_0400
Remote Control Receive Enable Register	RMCREN	0x4004_0404
Remote Control Receive Data Buffer Register 1	RMCRBUF1	0x4004_0408
Remote Control Receive Data Buffer Register 2	RMCRBUF2	0x4004_040C
Remote Control Receive Data Buffer Register 3	RMCRBUF3	0x4004_0410
Remote Control Receive Control Register 1	RMCRCR1	0x4004_0414
Remote Control Receive Control Register 2	RMCRCR2	0x4004_0418
Remote Control Receive Control Register 3	RMCRCR3	0x4004_041C
Remote Control Receive Control Register 4	RMCRCR4	0x4004_0420
Remote Control Receive Status Register	RMCRSTAT	0x4004_0424
Remote Control Receive End bit Number Register1	RMCEND1	0x4004_0428
Remote Control Receive End bit Number Register2	RMCEND2	0x4004_042C
Remote Control Receive End bit Number Register3	RMCEND3	0x4004_0430
Remote Control Source Clock selection Register	RMCFSSEL	0x4004_0434

16.2.2 Remote Control Enable Register [RMCEN]

	7	6	5	4	3	2	1	0
bit Symbol	—						—	RMCEN
Read/Write	R						R/W	R/W
After reset	0						0	0
Function	”0” is read.						Write as “1”.	RMC operation 0: Disable 1: Enable

<RMCEN>: Controls RMC operation.
To allow RMC to function, enable the RMCEN bit first. If the operation is disabled, all the clocks for RMC except for the enable register are stopped, and it can reduce power consumption.
If RMC is enabled and then disabled, the settings in each register remain intact.

16.2.3 Remote Control Receive Enable Register [RMCREN]

	7	6	5	4	3	2	1	0
bit Symbol	—							RMCREN
Read/Write	R							R/W
After reset	0							0
Function	”0” is read.							Reception 0: Disable 1: Enable

<RMCREN>: Controls reception of RMC.
Setting this bit to “1” enables reception.

(Note) Enable the <RMCREN> bit after setting the RMCxRCR1, RMCxRCR2 and RMCxRCR3.

16.2.4 Remote Control Receive Data Buffer Register 1 [RMCRBUF1]

	31	30	29	28	27	26	25	24
bit Symbol	RMCRBUF31	RMCRBUF30	RMCRBUF29	RMCRBUF28	RMCRBUF27	RMCRBUF26	RMCRBUF25	RMCRBUF24
Read/Write	R							
After reset	0							
Function	Received data							
	23	22	21	20	19	18	17	16
bit Symbol	RMCRBUF23	RMCRBUF22	RMCRBUF21	RMCRBUF20	RMCRBUF19	RMCRBUF18	RMCRBUF17	RMCRBUF16
Read/Write	R							
After reset	0							
Function	Received data							
	15	14	13	12	11	10	9	8
bit Symbol	RMCRBUF15	RMCRBUF14	RMCRBUF13	RMCRBUF12	RMCRBUF11	RMCRBUF10	RMCRBUF9	RMCRBUF8
Read/Write	R							
After reset	0							
Function	Received data							
	7	6	5	4	3	2	1	0
bit Symbol	RMCRBUF7	RMCRBUF6	RMCRBUF5	RMCRBUF4	RMCRBUF3	RMCRBUF2	RMCRBUF1	RMCRBUF0
Read/Write	R							
After reset	0							
Function	Received data							

<RMCRBUF31:0>: Reads 4 bytes of received data.

16.2.5 Remote Control Receive Data Buffer Register 2 [RMCRBUF2]

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

<RMCRBUF63:32>: Reads 4 bytes of received data.

16.2.6 Remote Control Receive Data Buffer Register 3 [RMCRBUF3]

	7	6	5	4	3	2	1	0
bit Symbol	RMCRBUF 71	RMCRBUF 70	RMCRBUF 69	RMCRBUF 68	RMCRBUF 67	RMCRBUF 66	RMCRBUF 65	RMCRBUF 64
Read/Write	R							
After reset	0							
Function	Received data							

<RMCRBUF71:64>: Reads a byte of received data.

(Note 1) **Received data is stored from RMCRBUF1 <RMCRBUF0> to RMCRBUF3 <RMCRBUF71> in sequence.**

(Note 2) **The first received bit is stored in the MSB. The last received bit is stored in the LSB (bit 0).**

If the remote control signal is received in the LSB first algorithm, the received data is stored in reverse sequence.

16.2.7 Remote Control Receive Control Register 1 [RMCRCR1]

	31	30	29	28	27	26	25	24
bit Symbol	RMCLC MAX7	RMCLC MAX6	RMCLC MAX5	RMCLC MAX4	RMCLC MAX3	RMCLC MAX2	RMCLC MAX1	RMCLC MAX0
Read/Write	R/W							
After reset	0							
Function	Maximum cycle of leader detection: RMCLCMAX×4/fs[s]							

	23	22	21	20	19	18	17	16
bit Symbol	RMCLC MIN7	RMCLC MIN6	RMCLC MIN5	RMCLC MIN4	RMCLC MIN3	RMCLC MIN2	RMCLC MIN1	RMCLC MIN0
Read/Write	R/W							
After reset	0							
Function	Minimum cycle of leader detection: RMCLCMIN×4/fs[s]							

	15	14	13	12	11	10	9	8
bit Symbol	RMCLL MAX7	RMCLL MAX6	RMCLL MAX5	RMCLL MAX4	RMCLL MAX3	RMCLL MAX2	RMCLL MAX1	RMCLL MAX0
Read/Write	R/W							
After reset	0							
Function	Maximum low width of leader detection: RMCLLMAX×4/fs[s]							

	7	6	5	4	3	2	1	0
bit Symbol	RMCLL MIN7	RMCLL MIN6	RMCLL MIN5	RMCLL MIN4	RMCLL MIN3	RMCLL MIN2	RMCLL MIN1	RMCLL MIN0
Read/Write	R/W							
After reset	0							
Function	Minimum low width of leader detection: RMCLLMIN×4/fs[s]							

<RMCLCMAX7:0>: Specifies a maximum cycle of leader detection.
Calculating formula of the maximum cycle: RMCLCMAX×4/fs[s].
RMC detects the first cycle as a leader if it is within the maximum cycle.

<RMCLCMIN7:0>: Specifies a minimum cycle of leader detection.
Calculating formula of the minimum cycle: RMCLCMIN×4/fs[s].
RMC detects the first cycle as a leader if it exceeds the minimum cycle.

<RMCLLMAX7:0>: Specifies a maximum low width of leader detection.
Calculating formula of the maximum low width: RMCLLMAX×4/fs[s]
RMC detects the first cycle as a leader if its low width is within the maximum low width.

<RMCLLMIN7:0>: Specifies a minimum low width of leader detection.
Calculating formula of the minimum low width: RMCLLMIN×4/fs[s]
RMC detects the first cycle as a leader if its low width exceeds the minimum low width.
If RMCRCR2<RMCLD> = 1, a value less than the specified is determined as data.

(Note) When you configure the register, you must follow the rule shown below.

Leader	Rules
Low width + high width	<RMCLCMAX7:0> > <RMCLCMIN7:0> <RMCLLMAX7:0> > <RMCLLMIN7:0> <RMCLCMIN7:0> > <RMCLLMAX7:0>
Only with high width	<RMCLCMAX7:0> > <RMCLCMIN7:0> <RMCLLMAX7:0> = 0y00000000 <RMCLLMIN7:0> = don't care
No leader	<RMCLCMAX7:0> = 0y00000000 <RMCLCMIN7:0> = don't care <RMCLLMAX7:0> = don't care <RMCLLMIN7:0> = don't care

16.2.8 Remote Control Receive Control Register 2 [RMCRCR2]

	31	30	29	28	27	26	25	24
bit Symbol	RMCLIEN	RMCEDIEN	—	—	—	—	RMCLD	RMCPHM
Read/Write	R/W	R/W	R				R/W	R/W
After reset	0	0	0				0	0
Function	Leader detection interrupt 0: Not generated 1 : Generated	Remote control input falling edge interrupt 0: Not generated 1 : Generated	”0” is read.				Receiving remote control signal with or without leader 0: Disable 1: Enable	Receive a remote control signal in phase method? 0: No (receive in cycle method) 1:Yes

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

	15	14	13	12	11	10	9	8
bit Symbol	RMCLL7	RMCLL6	RMCLL5	RMCLL4	RMCLL3	RMCLL2	RMCLL1	RMCLL0
Read/Write	R/W							
After reset	1							
Function	Excess low width that triggers reception completion and interrupt generation 0y00000000~0y11111110:RMCLL×1/fs[s] 0y11111111:not to use as the trigger							

	7	6	5	4	3	2	1	0
bit Symbol	RMCDMAX7	RMCDMAX6	RMCDMAX5	RMCDMAX4	RMCDMAX3	RMCDMAX2	RMCDMAX1	RMCDMAX0
Read/Write	R/W							
After reset	1							
Function	Maximum data bit cycle that triggers reception completion and interrupt generation 0y00000000~0y11111110:RMCDMAX×1/fs[s] 0y11111111: not to use as the trigger							

<RMCLIEN>: Enables to generate a leader detection interrupt by detecting a leader.

<RMCEDIEN>: Enables to generate a remote control input falling edge Interrupt.

<RMCLD>: Enables RMC to receive signals with or without a leader.

<RMCPHM>: Specifies data reception mode of a phase method. If you use the phase method of which signal cycle is fixed, set “1”.

<RMCLL7:0>: Specifies an excess low width. If an excess low width is detected, reception is completed and an interrupt is generated. The low width is not detected if <RMCLL7:0> = 0y11111111. Calculating formula of an excess low width: RMCLLx1/fs[s].

<RMCDMAX7:0>: Specifies a threshold for detecting a maximum data bit cycle. It is detected when a data bit cycle exceeds the threshold. It is not detected when <RMCMAX7:0> = 0y11111111. Calculating formula of the threshold: RMCDMAX x 1/fs[s].

16.2.9 Remote Control Receive Control Register 3 [RMCRCR3]

	15	14	13	12	11	10	9	8
bit Symbol	—	RMCDATH6	RMCDATH5	RMCDATH4	RMCDATH3	RMCDATH2	RMCDATH1	RMCDATH0
Read/Write	R	R/W						
After reset	0	0						
Function	”0” is read.	Larger threshold to determine a signal pattern in a phase method RMCDATH×1/fs[s]						
	7	6	5	4	3	2	1	0
bit Symbol	—	RMCDATL6	RMCDATL5	RMCDATL4	RMCDATL3	RMCDATL2	RMCDATL1	RMCDATL0
Read/Write	R	R/W						
After reset	0	0						
Function	”0” is read.	Threshold to determine 0 or 1 smaller threshold to determine a signal pattern in a phase method RMCDATL×1/fs[s]						

<RMCDATH6:0>: Specifies a larger threshold (within a range of 1.5T and 2T) to determine a pattern of remote control signal in a phase method. If the measured cycle exceeds the threshold, the bit is determined as “10”. If not, the bit is determined as “01”. Calculating formula of the threshold: RMCDATHx1/fs[s].

<RMCDATL6:0>: Specifies two kinds of thresholds: a threshold to determine whether a data bit is 0 or 1; a smaller threshold (within a range of 1T and 1.5T) to determine a pattern of remote control signal in a phase method.
As for the determination of data bit, if the measured cycle exceeds the threshold, the bit is determined as “1”. If not, the bit is determined as “0”. Calculating formula of the threshold: RMCDATL×1/fs[s].
As for the determination of a remote control signal pattern in a phase method, if the measured cycle exceeds the threshold, the bit is determined as “01”. If not, the bit is determined as “00”. Calculating formula of the threshold to determine 0 or 1: RMCDATLx1/fs[s].

(Note) If the <RMCPHM> bit of the Remote Control Receive Control Register 2 is “0”, <RMCDATH6:0> are not enabled. The bits are enabled when <RMCPHM> is ”1”.

16.2.10 Remote Control Receive Control Register 4 [RMCRCR4]

	7	6	5	4	3	2	1	0
bit Symbol	RMCPO	—	—	—	RMCNC3	RMCNC2	RMCNC1	RMCNC0
Read/Write	R/W	R			R/W			
After reset	0	0			0			
Function	Remote control input signal 0: Not reversed 1: Reversed	”0” is read.			Noise cancellation time 0000: No cancellation 0001~1111:RMCNC×1/fs[s]			

<RMCPO>: Specifies whether a remote control input signal is reversed or not.

<RMCNC3:0>: Specifies time noises are cancelled by a noise canceller. If <RMCNC3:0> = “0000”, noises are not cancelled. Calculating formula of noise cancellation time: RMCNC x 1/fs[s].

16.2.11 Remote Control Receive Status Register [RMCRSTAT]

	15	14	13	12	11	10	9	8
bit Symbol	RMCRLIF	RMCLOIF	RMCDMAXIF	RMCEDIF	—	—	—	—
Read/Write	R	R	R	R	R			
After reset	0	0	0	0	0			
	Leader detection is interrupt factor? 0: No 1: Yes	Low width detection is interrupt factor? 0: No 1: Yes	Maximum data bit cycle detection is interrupt factor? 0: No 1: Yes	Remote control input falling edge interrupt is interrupt factor? 0: No 1: Yes	"0" is read.			
	7	6	5	4	3	2	1	0
bit Symbol	RMCRLDR	RMCRNUM6	RMCRNUM5	RMCRNUM4	RMCRNUM3	RMCRNUM2	RMCRNUM1	RMCRNUM0
Read/Write	R	R						
After reset	0	0						
Function	Leader detection 0: No 1: Yes	The number of received data bit 0y0000000:no data bit (only with leader) 0y0000001~0y1001000:1~72bit 0y1001001~0y1111111:73bit and more						

<RMCRLIF>: Indicates that leader detection is the interrupt factor.

<RMCLOIF>: Indicates that low width detection is the interrupt factor.

<RMCDMAXIF>: Indicates that maximum data bit cycle detection is the interrupt factor.

<RMCEDIF>: Indicates that a remote control input falling edge interrupt is the interrupt factor.

<RMCRLDR>: Detects a leader of a received remote control signal

<RMCRNUM6:0>: Indicates the number of bits received as remote control signal data. The number cannot be monitored during reception. On completion of reception, the number is stored.

(Note 1) This register is updated every time an interrupt is generated.
Writing to this register is ignored.

(Note 2) RMC keeps receiving 73 bit or more data unless reception is completed by detecting the maximum data bit cycle or the excess low width. If so, the received data in the data buffer may not be correct.

16.2.12 Remote Control Receive End Bit Number Register 1 [RMCEND1]

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCEND1						
Read/Write	R	R/W						
After reset	0	0						
Function	”0” is read.	Specifies that the number of receive data bit 0000000: No specifically the receive data bit 0000001 to 1001000: 1bit to 72bit 1001001 to 1111111: Don’t set the value						

<RMCEND1>: Specifies that the number of receive data bit

16.2.13 Remote Control Receive End Bit Number Register 2 [RMCEND2]

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCEND2						
Read/Write	R	R/W						
After reset	0	0						
Function	”0” is read.	Specifies that the number of receive data bit 0000000: No specifically the receive data bit 0000001 to 1001000: 1bit to 72bit 1001001 to 1111111: Don’t set the value						

<RMCEND2>: Specifies that the number of receive data bit

16.2.14 Remote Control Receive End Bit Number Register 3 [RMCEND3]

	7	6	5	4	3	2	1	0
bit Symbol	—	RMCEND3						
Read/Write	R	R/W						
After reset	0	0						
Function	”0” is read.	Specifies that the number of receive data bit 0000000: No specifically the receive data bit 0000001 to 1001000: 1bit to 72bit 1001001 to 1111111: Don’t set the value						

<RMCEND3>: Specifies that the number of receive data bit

(Note 1) As specified to RMCEND[3:1], it is able to set three kinds of the receive data bit.
(Note 2) To use the RMCEND[3:1] is in combination with the maximum data bit cycle.

16.2.15Remote Control Source Clock selection Register [RMCFSSEL]

	7	6	5	4	3	2	1	0
bit Symbol	—							RMCCLK
Read/Write	R							R/W
After reset	0							0
Function	”0” is read.							RMC Sampling clock 0：Low frequency Clock (32kHz) 1：TB1OUT

<RMCCLK>: Specifies that Sampling clock of RMC function

For the Sampling of RMC function ,It is able to set the Low Frequency Clock(32kHz) or Timer output （TB1OUT）.

The Setting range of Timer output by TB1OUT is from 30kHz to 34kHz.

（Note） To Change the sampling clock by RMCFSSEL, at first Stopping RMC operation by RMCEN<bit0> ,
After resetting enable operation , set the RMCFSSEL before the other RMC registers.

16.3 Operation Description

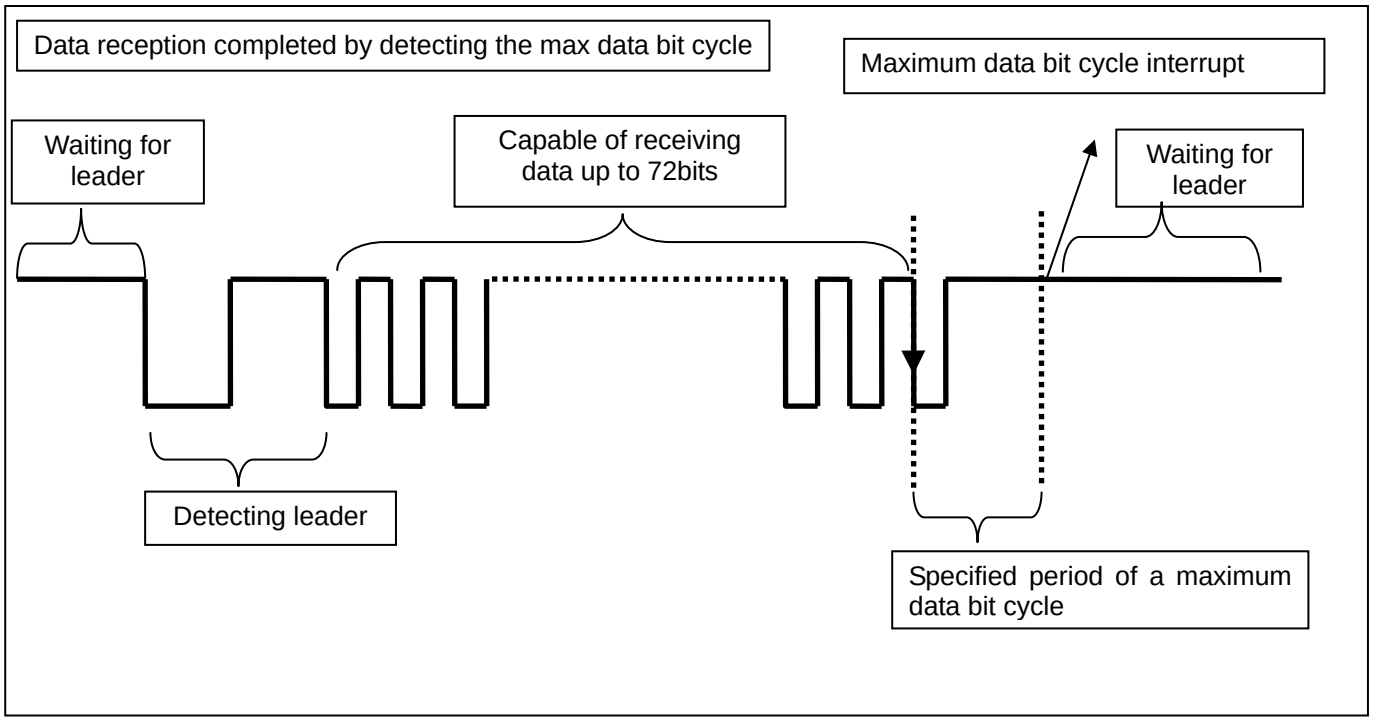
16.3.1 Reception of Remote Control Signal

16.3.1.1 Sampling Clock

A remote control signal is sampled by low-speed clock (fs).

16.3.1.2 Basic Operation

RMC starts to receive a data bit if a leader is detected while RMC is waiting for a leader. Based on a falling edge cycle, the data bit is determined as 0 or 1. By detecting a leader while RMC is waiting for a leader, a leader detection interrupt is generated, and the data bit reception starts. The data bit is determined as 0 or 1 based on a falling edge cycle. RMC is capable of receiving data up to 72bit. Reception is completed by detecting either a maximum data bit cycle or the excess low width. On completion of reception, RMC is waiting for the next leader, and the Remote Control Receive Data Buffer Registers and the Remote Control Receive Status Register are updated.



16.3.1.3 Preparation

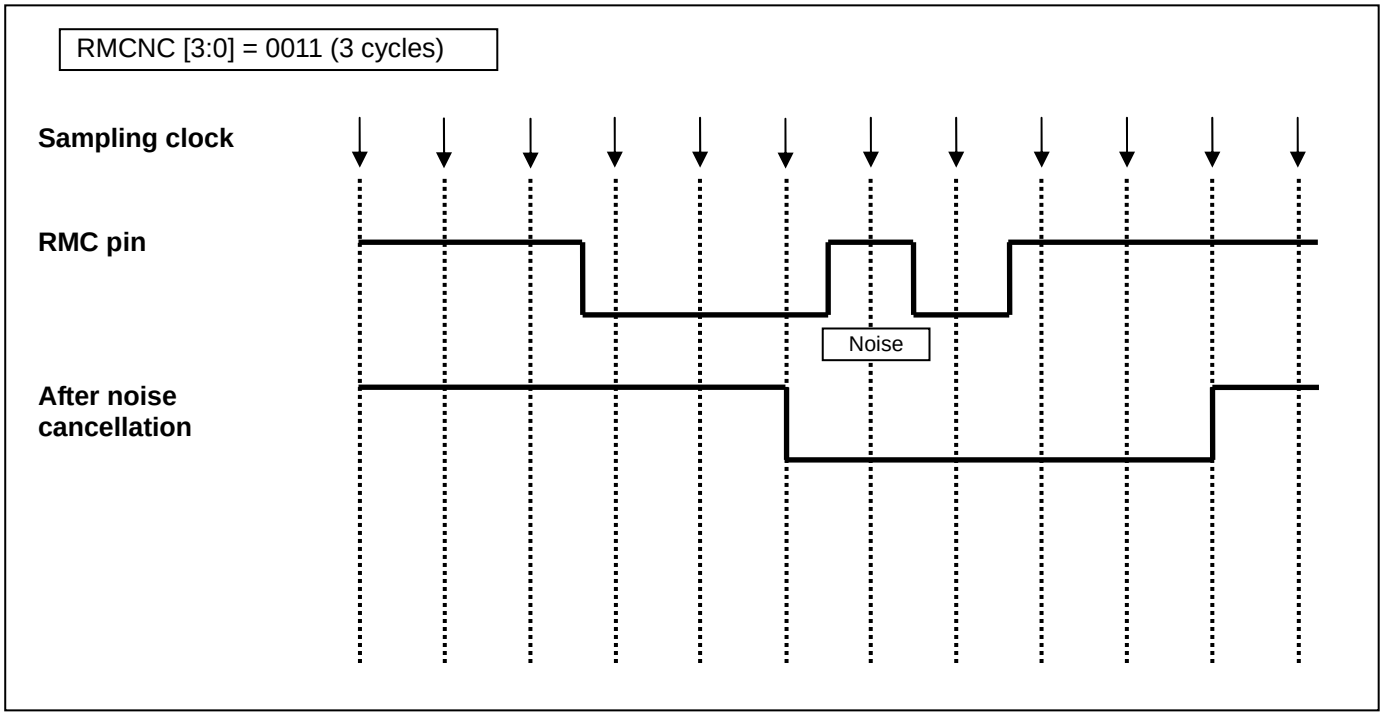
Configure reception operation of a remote control signal with the Remote Control Signal Receive Control Registers (RMCRCR1, RMCRCR 2 and RMCRCR3) before reception.

(1) Settings of Noise Cancelling Time

Configure noise cancelling time with the RMCRCR4 <RMCNC3:0> bit.

RMC monitors a remote control signal in each rising edge of a sampling clock. If “1” is monitored, RMC recognizes that the signal was changed to “0” after monitoring cycles of “0”s specified in RMCNC. If “0” is monitored, RMC recognizes that the signal was changed to “1” after monitoring cycles of “1”s specified in RMCNC.

The following figure shows how RMC operates according to the noise cancel setting of RMCNC [3:0] = 0011 (3 cycles). Subsequent to noise cancellation, the signal is changed from “1” to “0” upon monitoring 3 cycles of “0” s, and the signal is changed from “0” to “1” upon monitoring 3 cycles of “1” s.

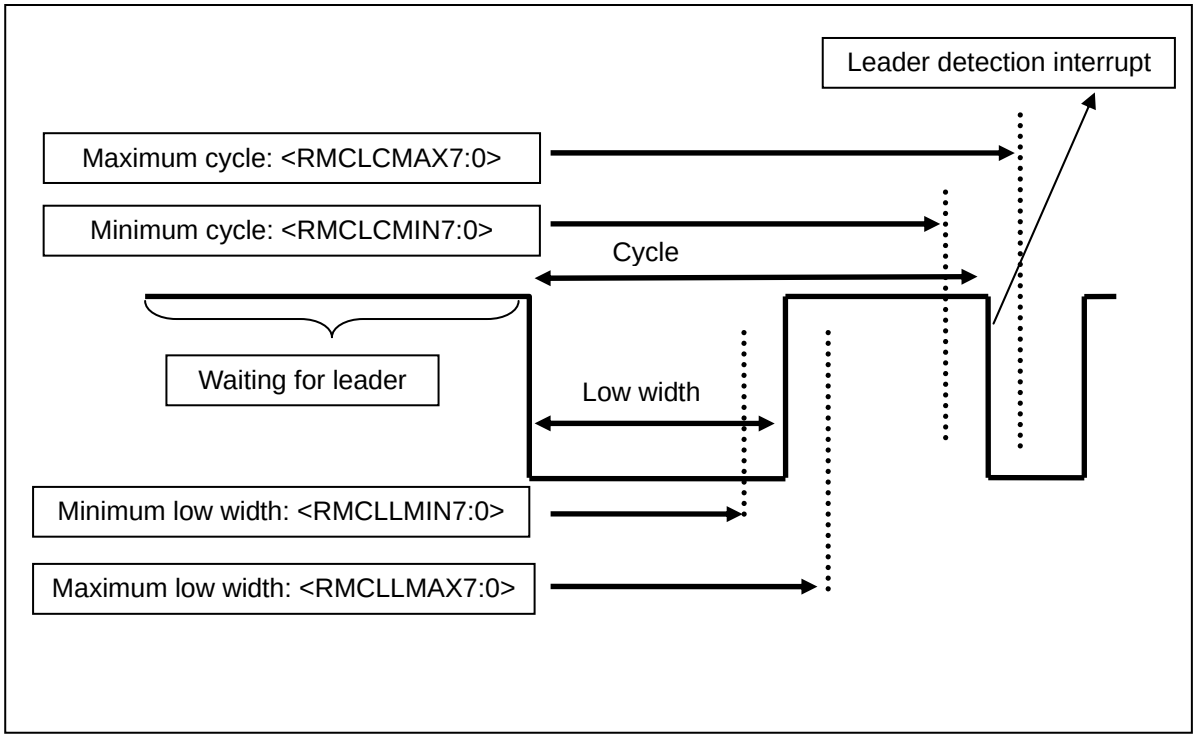


(2) Settings of Detecting Leader

To detect a leader, configure cycle and low width of the leader with the RMCRCR1 <RMCLLMIN7:0> <RMCLLMAX7:0> <RMCLCMIN7:0> <RMCLCMAX7:0> bits. When you configure the register, you must follow the rule shown below.

Leader	Rules
Low width + High width	<RMCLCMAX7:0> > <RMCLCMIN7:0> <RMCLLMAX7:0> > <RMCLLMIN7:0> <RMCLCMIN7:0> > <RMCLLMAX7:0>
Only with high width	<RMCLCMAX7:0> > <RMCLCMIN7:0> <RMCLLMAX7:0> = 0y00000000 <RMCLLMIN7:0> = don't care
No leader	<RMCLCMAX7:0> = 0y00000000 <RMCLCMIN7:0> = don't care <RMCLLMAX7:0> = don't care <RMCLLMIN7:0> = don't care

The following shows a leader waveform and the RMCRCR1 register settings.



If you want to generate an interrupt when detecting a leader, configure the RMCRCR2 <RMCLIEN> bit. A remote control signal without a leader cannot generate a leader detection interrupt.

(3) Settings of Data Bit Determination

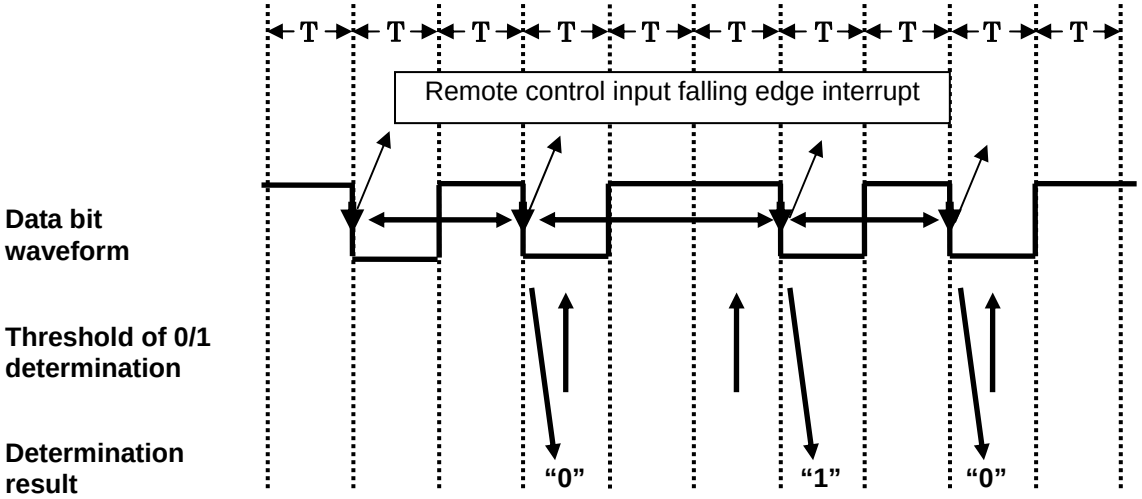
Based on a falling edge cycle, the data bit is determined as 0 or 1.

Configure a threshold of the determination with the RMCRCR3 <RMCDATL6:0> bit. If the cycle exceeds the threshold, the bit is determined as “1”. If not, the bit is determined as “0”.

By setting ”1” to the RMCRCR2 <RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of the data bit. Using this interrupt together with a 16-bit timer enables the determination to be done by software.

The following shows how the data bit is determined as “0” or “1”.

Example: Threshold of 0/1 determination is set to 2.5T with the <RMCDATL6:0> bit.



As for data bit determination of a remote control signal in a phase method, see 16.3.1.10 “Receiving a Remote Control Signal in a Phase Method”.

(4) Settings of Reception Completion

To complete data reception, settings of detecting the maximum data bit cycle and excess low width are required. If multiple factors are specified, reception is completed by the factor detected first. Make sure to configure the reception completion settings.

1) Completed by a maximum data bit cycle

To complete reception by detecting a maximum data bit cycle, you need to configure the RMCRCR2 <RMCDMAX7:0> bits. If the falling edge of the data bit cycle isn't monitored after time specified as threshold in the <RMCDMAX7:0> bits, a maximum data bit cycle is detected. The detection completes reception and generates an interrupt.

To complete reception by setting the number of receive data is set a RMCEND 1 to 3 register Of each <RMCEND1>、<RMCEND2>、<RMCEND3>.

In this case when the number of set reception bit agreed with the number of bit which I received at the time of the outbreak of MAX on <RMCEND1> of three RMCEND1 - registers, <RMCEND2>, <RMCEND3> in data bit period, it occurs by an MAX interrupt in data bit period.

To set the number of receive data can be set in the three RMCEND1>、<RMCEND2>、<RMCEND3> register.

When it can receive the Maximum Data bit , the number of bit is not match the setting value in <RMCEND1>、<RMCEND2>、<RMCEND3> register., it wait for Leader Reception.

2) Completed by excess low width

To complete reception by detecting the low width, you need to configure the RMCRCR2 <RMCLL7:0> bits. After the falling edge of the data bit is detected, if the signal stays low longer than specified, excess low width is detected. The detection completes reception and generates an interrupt.

16.3.1.4 Enabling Reception

By enabling the RMCREN <RMCREN> bit after configuring the RMCRCR1, RMCRCR2, RMCRCR3 and RMCRCR4 registers, RMC is ready for reception. Detecting a leader initiates reception.

(Note) Changing the configurations of the RMCRCR1, RMCRCR2, RMCRCR3 and RMCRCR4 registers during reception may harm their proper operation. Be careful if you change them during reception.

16.3.1.5 Reception

Detecting a leader sets the RMCRSTAT <RMCRLDR> bit. Simultaneously, a leader detection interrupt is generated if the RMCRCR2 <RMCLIEN> bit is set. When the interrupt is generated, the RMCRSTAT <RMCRLIF> bit is set.

Next to the leader detection, each data bit is determined as 0 or 1. The results are stored in the RMCRBUF1, RMCRBUF 2 and RMCRBUF 3 registers up to 72bits. By setting “1” to the RMCRCR2 <RMCEDIEN> bit, a remote control signal input falling edge interrupt can be generated in each falling edge of the data bit. When the interrupt is generated, the RMCRSTAT <RMCEDIF> bit is set.

Detecting the maximum data bit cycle or the excess low width completes reception and generates an interrupt. Only when the received number of bit until detecting data bit cycle MAX is corresponding, it becomes reception end/interruption generation when <RMCEND1>, <RMCEND2>, and <RMCEND3> of the RMCEND1 to 3 register are set.

To check the status of RMC after reception is completed, read the Remote Control Receive Status Register.

On completion of reception, RMC is waiting for the next leader.

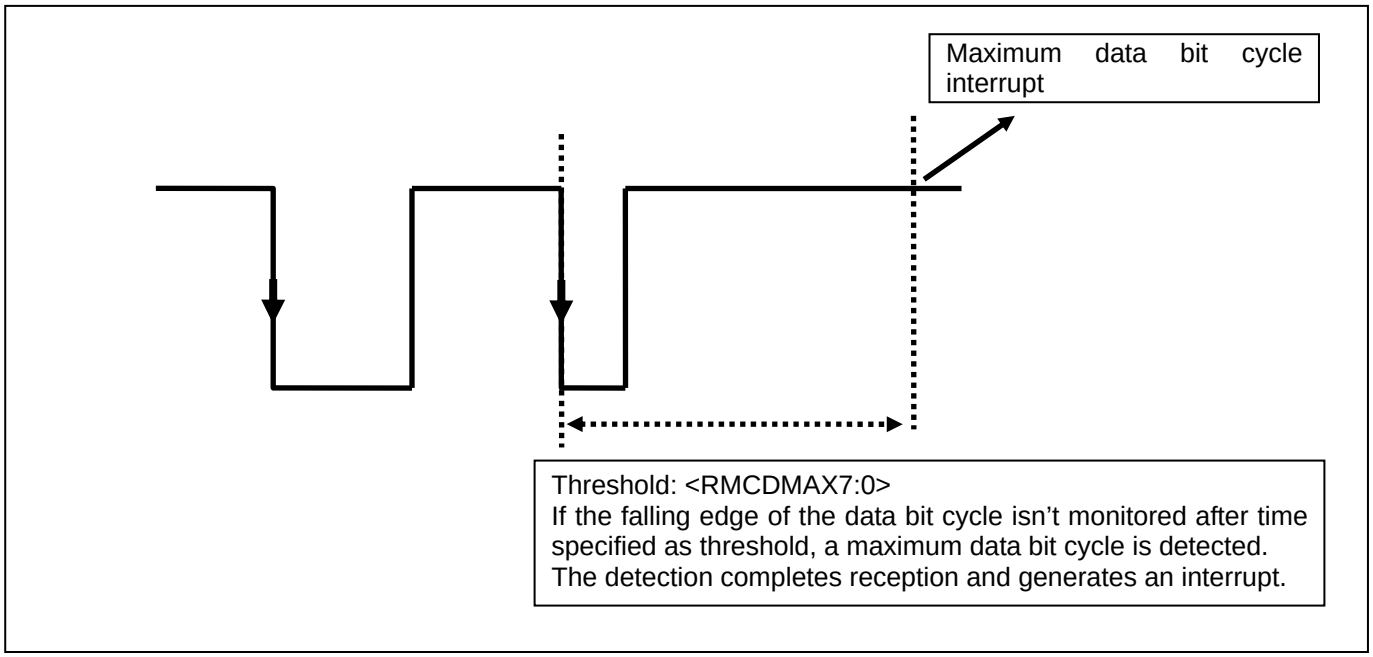
By setting RMC to receive a signal without a leader, RMC recognizes the received is data and starts reception without detecting a leader.

If the next data reception is completed before reading the preceding received data, the preceding data is over-written by the next one.

16.3.1.6 Reception Completion

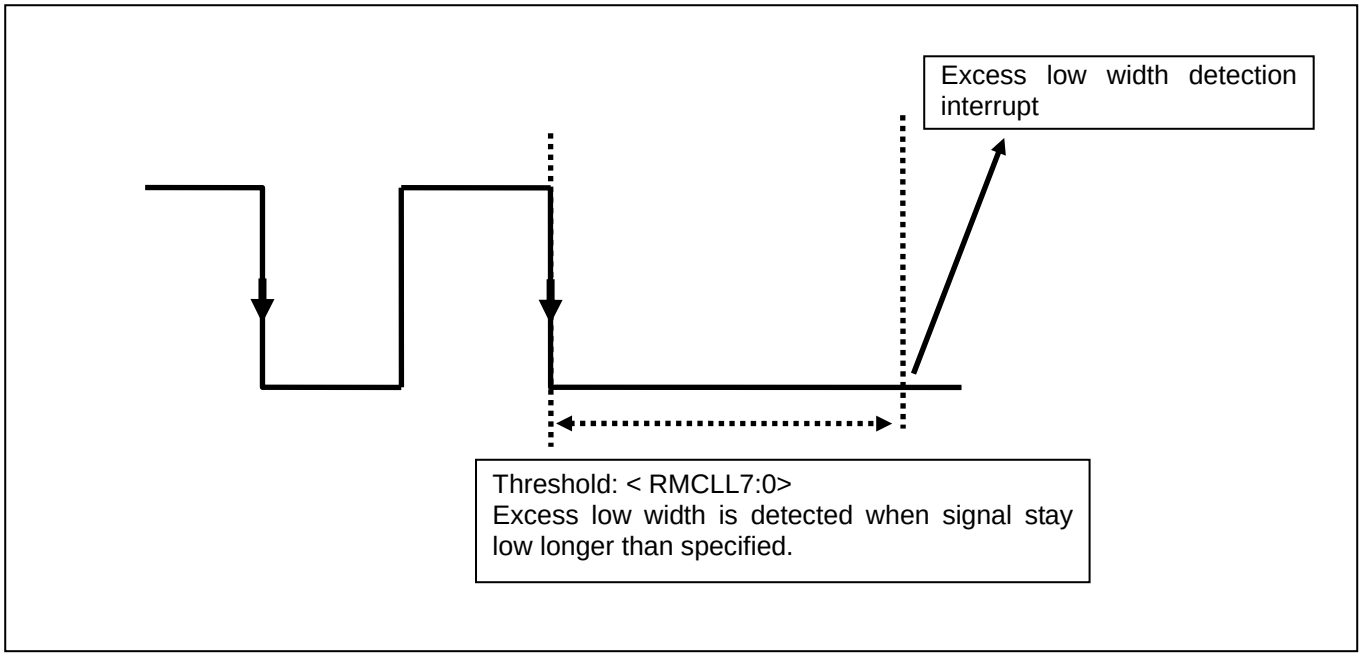
1) Completed by a maximum data bit cycle

Detecting a maximum data bit cycle completes reception and generates an interrupt. After the interrupt is generated, the RMCRSTAT <RMCDMAXIF> bit is set to “1”.



2) Completed by excess low width

Detecting excess low width completes reception and generates an interrupt. After the interrupt is generated, the RMCRSTAT <RMCLOIF> bit is set to “1”.



RMC keeps receiving 73 bit or more data unless reception is completed by detecting the maximum data bit cycle or the excess low width. If so, the received data in the data buffer may not be correct.

To check the status of RMC after reception is completed, read the Remote Control Receive Status Register. The status of RMC that each bit type indicates is shown below.

<RMCRLDR>	<RMCRNUM6:0>	RMC Status
0	0000001~1001000	Receiving remote control signal without a leader (Data bits: 1~72bit)
0	1001001~1111111	Receiving remote control signal without a leader (Data bits: 73bit and more)
1	0000000	Only with a leader
1	0000001~1001000	Receiving remote control signal with a leader (Data bits: 1~72bit)
1	1001001~1111111	Receiving remote control signal without a leader (Data bits: 73bit and more)

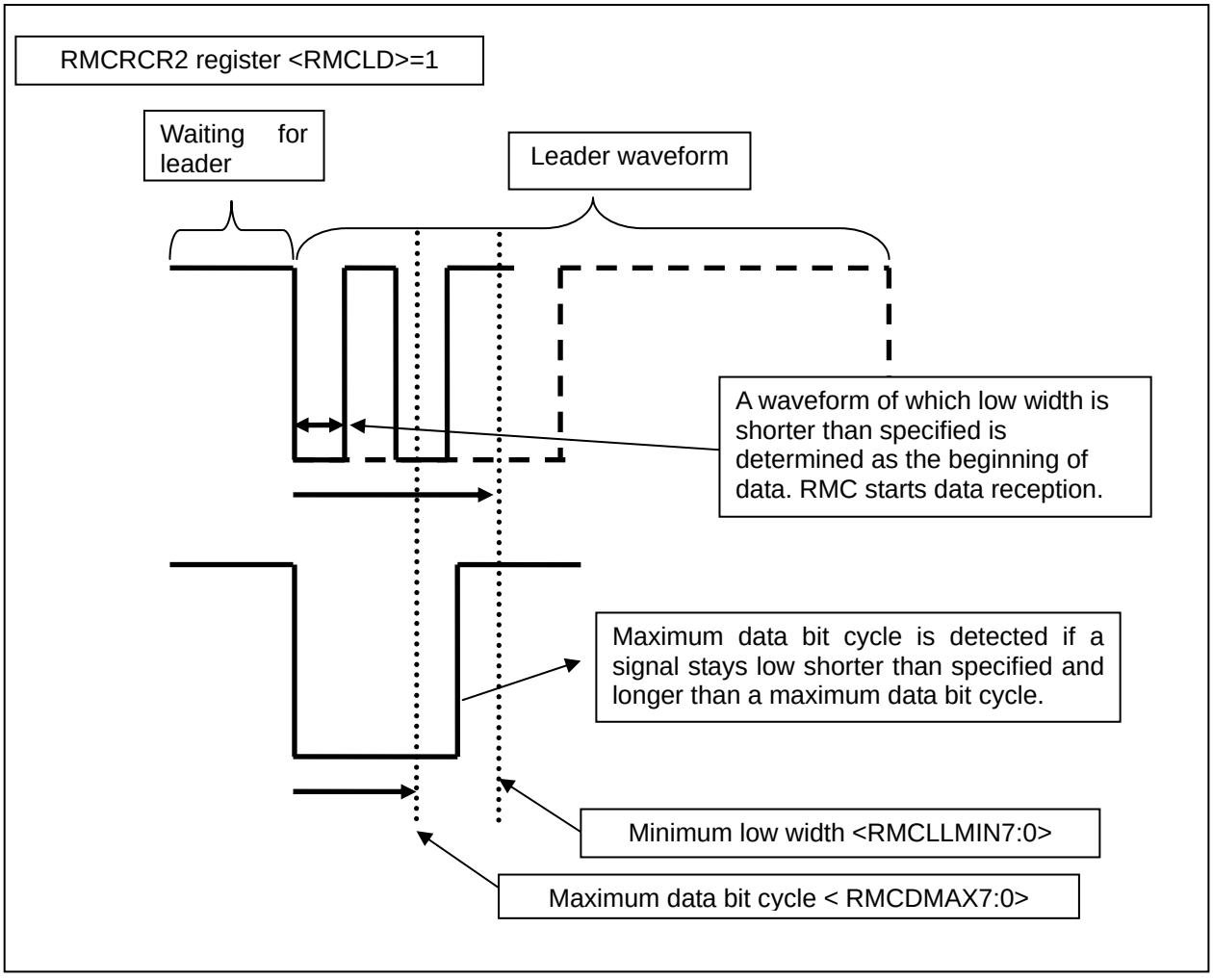
16.3.1.7 Stopping Reception

RMC stops reception by clearing the RMCREN <RMCREN> bit to “0” (reception disabled). Clearing this bit during reception stops reception immediately and the received data is discarded.

16.3.1.8 Receiving Remote Control Signal without Leader

Setting RMCRCR2 <RMCLD> enables RMC to receive signals with or without a leader. By setting RMCRCR2 <RMCLD>, RMC starts receiving data if it recognizes a signal of which low width is shorter than a maximum low width of leader detection specified in the RMCRCR1 <RMCLLMAX7:0> bits. RMC keeps receiving data until the final data bit is received.

If RMCRCR2 <RMCLD> is enabled, the same settings of error detection, reception completion and data bit determination of 0 or 1 are applied regardless of whether a signal has a leader or not. Thus receivable remote control signals are limited.



16.3.1.9 A Leader only with Low Width

The figure shown below illustrates a remote control signal that starts with a leader of which waveform only has low width. This signal starts with a leader that only has low width and a data bit cycle starts from the rising edge. To enable the signal, it must be sent after being reversed by setting the RMCRCR4 <RMCPO> bit to “1”. This is because RMC is configured to detect a data bit cycle from the falling edge.

A leader is detected by the low width. When you configure the RMCRCR1 register, you must follow the rule shown below.

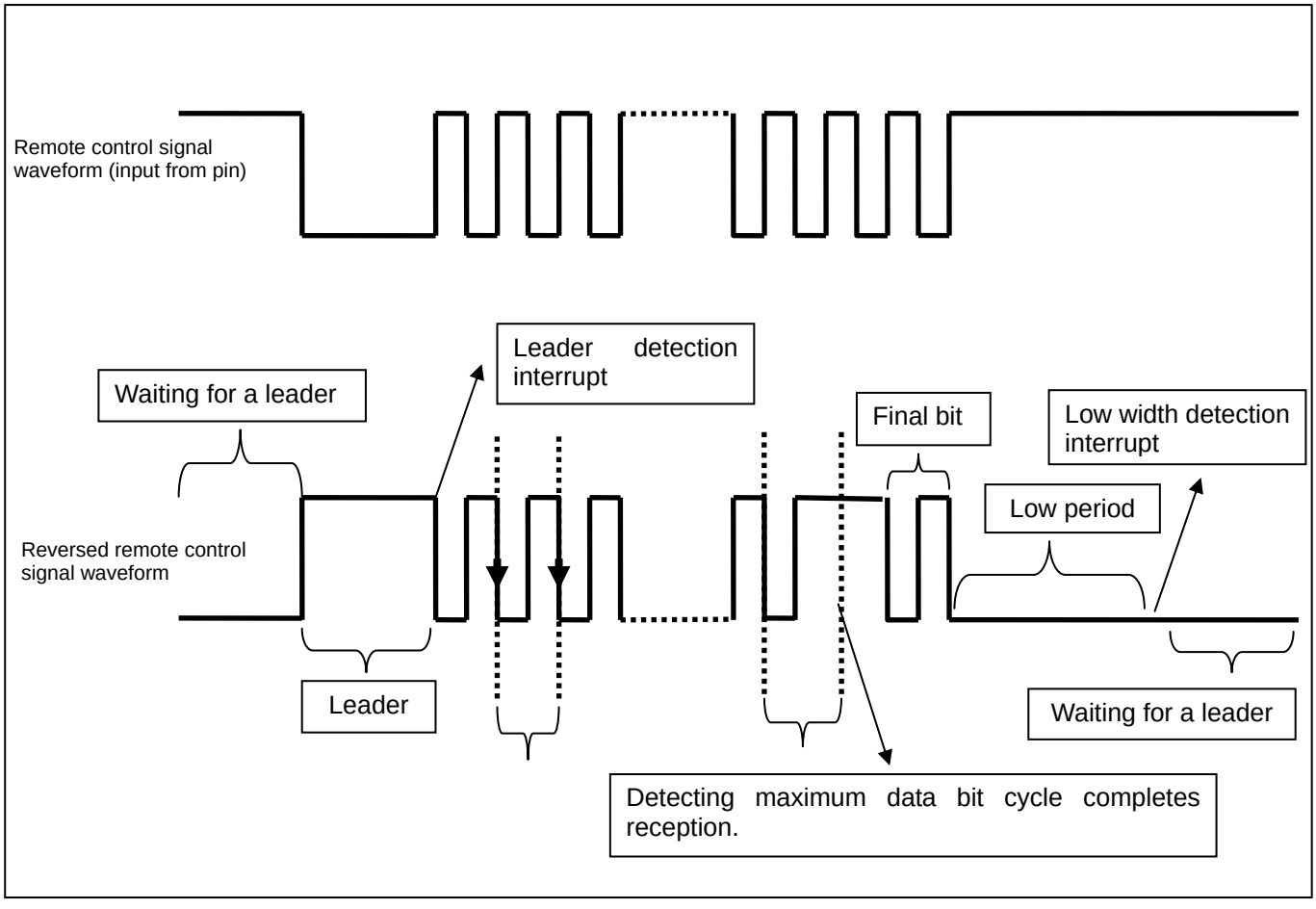
<RMCLLMAX7:0> = 0y00000000
<RMCLCMAX7:0> > <RMCLCMIN7:0>

If the rules are applied, RMC does not care about the value of <RMCLLMIN7:0>.

To determine the data bit as 0 or 1, configure a threshold of the determination with the RMCRCR3 <RMCDATL6:0> bit.

Configure a maximum data bit cycle with the <RMCDMAX7:0> bits of the Remote Control Receive Control Register 2.

To complete reception by detecting the maximum data bit cycle, you need to configure the RMCRCR2 <RMCDMAX7:0> bits. To complete reception by detecting the low width, you need to configure the RMCRCR2 <RMCLL7:0> bits. Detecting the maximum data bit cycle or the excess low width completes reception and generates an interrupt. RMC waits for the next leader.



16.3.1.10 Receiving a Remote Control Signal in a Phase Method

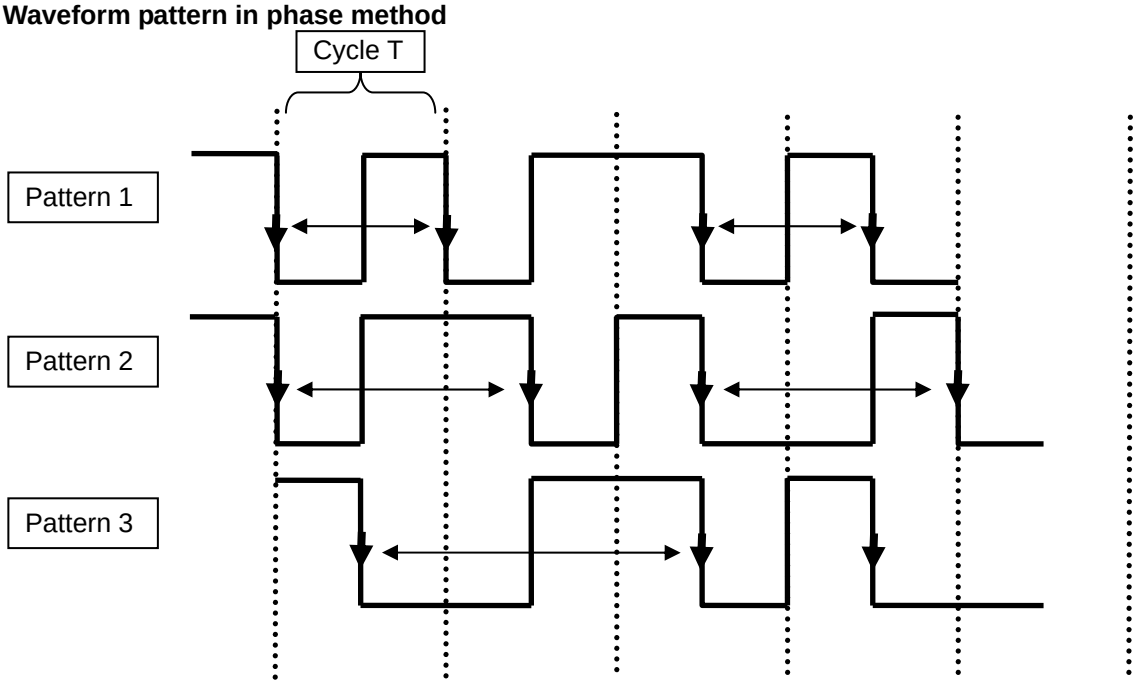
RMC is capable of receiving a remote control signal in a phase method of which signal cycle is fixed. A signal in the phase method has three waveform patterns (see the figure shown below). By setting two thresholds a remote control signal pattern is determined. RMC converts the signal into data ”0” or “1”. On completion of reception, received data “0” and “1” are stored in the RMCRBUF1, RMCRBUF 2 and RMCRBUF3.

By setting RMCRCR2<RMCPHM>=”1”, RMC enables to receive a remote control signal in the phase method. Each threshold can be configured with the RMCRCR3 <RMCDATL6:0> bits and <RMCDATH6:0> bits. Two thresholds are used to distinguish three waveform patterns. On condition that a cycle between two falling edges is “T”, three patterns show cycles of 1T, 1.5T and 2T. Details of the two thresholds are shown below.

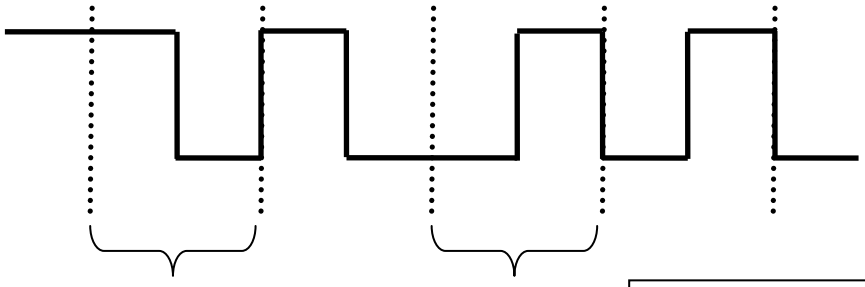
To determine a remote control signal in the phase method, three patterns of data waveform and preceding data are required. In addition, the signal needs to start from data “1”.

	Determined by	Threshold	Register bits to set
Threshold 1	Pattern 1 & pattern 2	1T~1.5T	RMCRCR3 register <RMCDATL6:0>
Threshold 2	Pattern 2 & pattern 3	1.5T~2T	RMCRCR3 register <RMCDATH6:0>

Waveform pattern in phase method

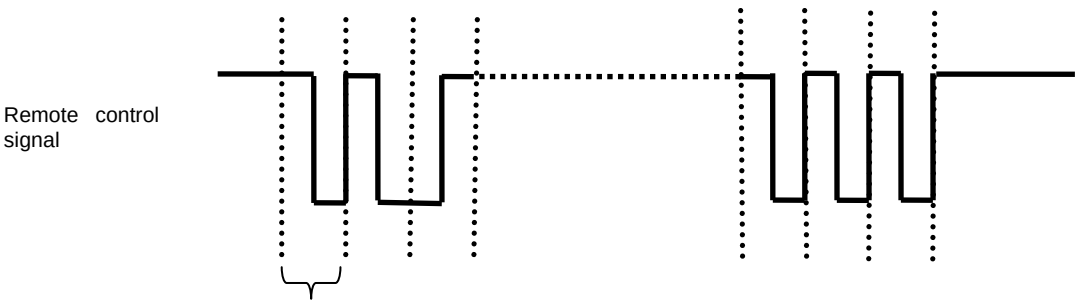


Remote control signal data in phase method



A pulse shape in a cycle indicates whether it is data “0” or data ”1”.

Remote control signal in phase method



The first two bits of data need to be “11”.

17 Watchdog Timer (WDT)

The watchdog timer (WDT) is for detecting malfunctions (runaways) of the CPU caused by noises or other disturbances and remedying them to return the CPU to normal operation. If the timer detects a runaway, it generates a non-maskable interrupt or an internal reset to notify the CPU. The watchdog timer starts immediately after reset release.

Note : INTWDT interrupt is a factor of the non-maskable interrpts(NMI).

17.1 Configuration

Fig 17-1 shows the block diagram of the watchdog timer

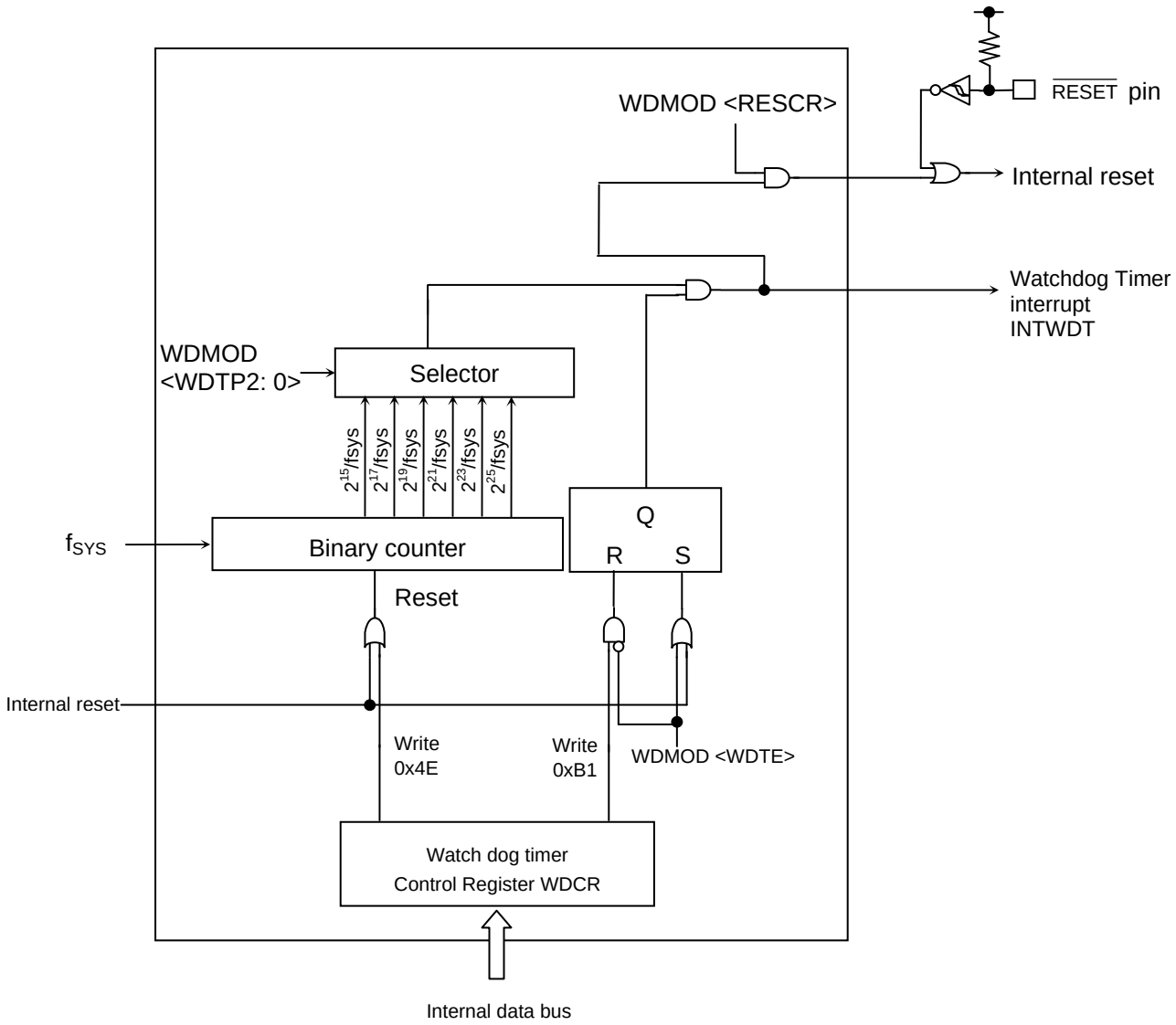


Fig 17-1 Block Diagram of the Watchdog Timer

17.2 Outline

The watchdog timer consists of the binary counters that are arranged in 25 stages and work using the f_{SYS} system clock as an input clock. The outputs produced by these binary counters are 2^{15}, 2^{17}, 2^{19}, 2^{21}, 2^{23} are 2^{25}.By selecting one of these outputs with WDMOD <WDTP2:0>, INTWDT can be generated when an overflow occurs, as shown in Fig 17-2.

17.2.1 INTWDT (WDMOD<RESCR>=0)

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

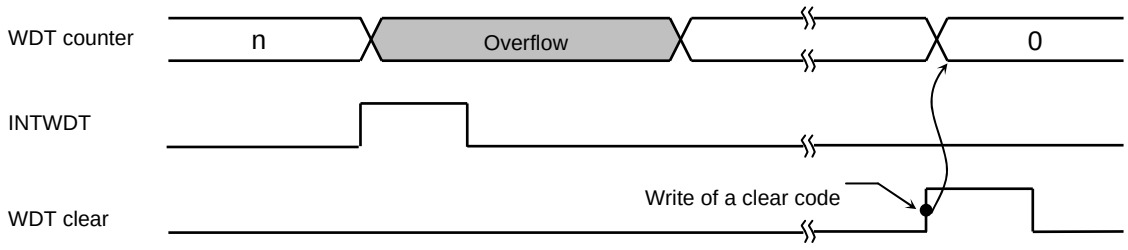


Fig 17-2 Normal Mode

17.2.2 Reset Mode(WDMOD<RESCR>=1)

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is affected for a 32-state time, as shown in Fig 17-3. If this reset is affected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 1 is used as an input clock f_{SYS}.

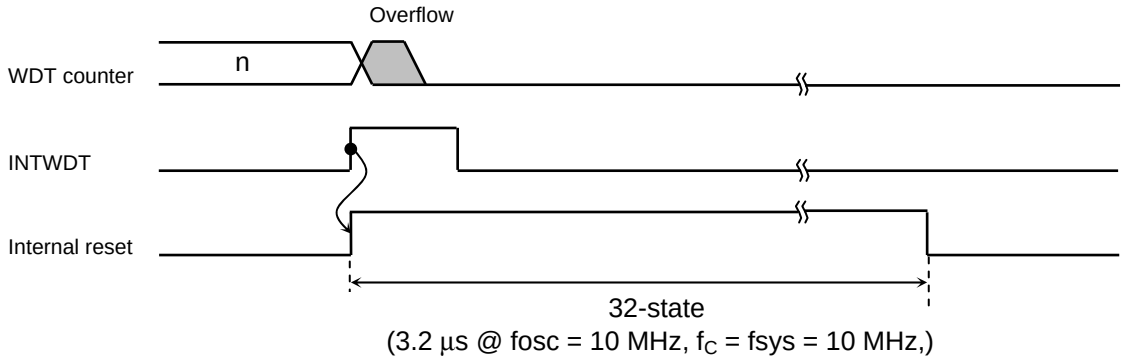


Fig 17-3 Reset Mode

17.3 Control Registers

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

17.3.1 Watchdog Timer Mode Register (WDMOD)

1. Enabling/disabling the watchdog timer <WDTE>

When resetting, WDMOD <WDTE> is initialized to "1" and the watchdog timer is enabled.

To disable the watchdog timer, this bit must be set to "0" and, at the same time, the disable code (0xB1) must be written to the WDCR register. This dual setting is intended to minimize the probability that the watchdog timer may inadvertently be disabled if a runaway occurs.

To change the status of the watchdog timer from "disable" to "enable," set the <WDTE> bit to "1".

2. Specifying the detection time of the watchdog timer <WDTP2: 0>

This is a 3-bit register for specifying the INTWDT time for runaway detection. When a reset is effected, this register is initialized to WDMOD <WDTP2: 0> = "000." Fig 17-4 shows the detection time of the watchdog timer.

3. Enabling/disabling the watchdog timer in IDLE mode <I2WDT>

Enabling/disabling the watchdog timer in IDLE mode is controlled by this bit. Writing “1” to this bit enables the watchdog timer and writing “0” to this bit disables the watchdog timer in IDLE mode.

(Note) Watchdog timer is stopped in stop mode.

4. Watchdog timer out reset connection <RESCR>

Setting this bit to "1" enables the watch dog timer to be reset when a runaway is detected. Since a reset initializes this bit to "1," a counter overflow causes a reset.

17.3.2 Watchdog Timer Control Register (WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

- Disabling control

By writing the disable code (0xB1) to this WDCR register after setting WDMOD <WDTE> to "0," the watchdog timer can be disabled.

```
WDMOD  ← 0 – – – – – – –   Clears WDTE to "0."
WDCR   ← 1 0 1 1 0 0 0 1   Writes the disable code (0xB1).
```

- Enabling control

Set WDMOD <WDTE> to "1".

- Watchdog timer clearing control

  Writing the clear code (0x4E) to the WDCR register clears the binary counter and allows it to resume counting.

  WDCR ← 0 1 0 0 1 1 1 0 Writes the clear code (0x4E)

(Note) Writing the disable code (0xB1) clears the binary counter.

Watchdog Timer Mode Register

WDMOD (0x4004_0000)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP2	WDTP1	WDTP0		I2WDT	RESCR	
Read/Write	R/W	R/W			R	R/W		R/W
After reset	1	0	0	0		0	1	0
Function	WDT control 0: disable 1: enable	Selects WDT detection time 000: $2^{15}/f_{SYS}$ 001: $2^{17}/f_{SYS}$ 010: $2^{19}/f_{SYS}$ 011: $2^{21}/f_{SYS}$ 100: $2^{23}/f_{SYS}$ 101: $2^{25}/f_{SYS}$ 110: Setting prohibited 111: Setting prohibited			"0" is read.	IDLE 0: Stop 1: Start	WDT out control 0: Generates INTWDT 1: Generates reset	Write "0."

Watchdog timer out control

0	Generates INTWDT
1	Generates reset

Detection time of watchdog timer @ fc =40 MHz

SYSCR1 clock gear value <GEAR2:0>	Detection time of watchdog timer WDMOD<WDTP2:0>					
	000	001	010	011	100	101
000 (fc)	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms
100 (fc/2)	1.63 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
101 (fc/4)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
110 (fc/8)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s
111 (fc/16)	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s	13.42 s

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Watchdog Timer Control Register

WDCR (0x4004_0004)

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

Disable & clear of WDT

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

Fig 17-4 Watchdog Timer Registers

17.4 Operation

The watchdog timer generates the INTWDT or an internal reset after a lapse of the detection time specified by the WDMOD <WDTP2: 0> register. Before generating the INTWDT or an internal reset, the binary counter for the watchdog timer must be cleared to "0" using software (instruction). If the CPU malfunctions (runaways) due to noise or other disturbances and cannot execute the instruction to clear the binary counter, the binary counter overflows and the non-maskable interrupt by the INTWDT or an internal reset is generated. The CPU is able to recognize the occurrence of a malfunction (runaway) by identifying the non-maskable interrupt and to restore the faulty condition to normal by using a malfunction (runaway) countermeasure program.

The watchdog timer begins operation immediately after a reset is cleared.

In STOP mode, the watchdog timer is reset and in an idle state. In IDLE mode, its operation depends on the WDMOD <I2WDT> setting. Before putting it in IDLE mode, WDMOD <I2WDT> must be set to an appropriate setting, as required.

Example:

1. To clear the binary counter

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Writes the clear code (0x4E)

2. To set the detection time of the watchdog timer to $2^{17}/f_{SYS.}$

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	0	1	–	–	–	–

3. To disable the watchdog timer.

		7	6	5	4	3	2	1	0	
WDMOD	←	0	–	–	–	–	–	–	–	Clears WDTE to "0"
WDCR	←	1	0	1	1	0	0	0	1	Writes the disable code (0xB1)

(Note 1) The counter of the watchdog timer stops at the debug mode.

18 Real Time Clock (RTC)

18.1 Functions

1) Clock (hour, minute and second)
2) Calendar (month, week, date and leap year)
3) Selectable 12 (am/ pm) and 24 hour display
4) Time adjustment + or − 30 seconds (by software)
5) Alarm (alarm output)
6) Alarm interrupt

18.2 Block Diagram

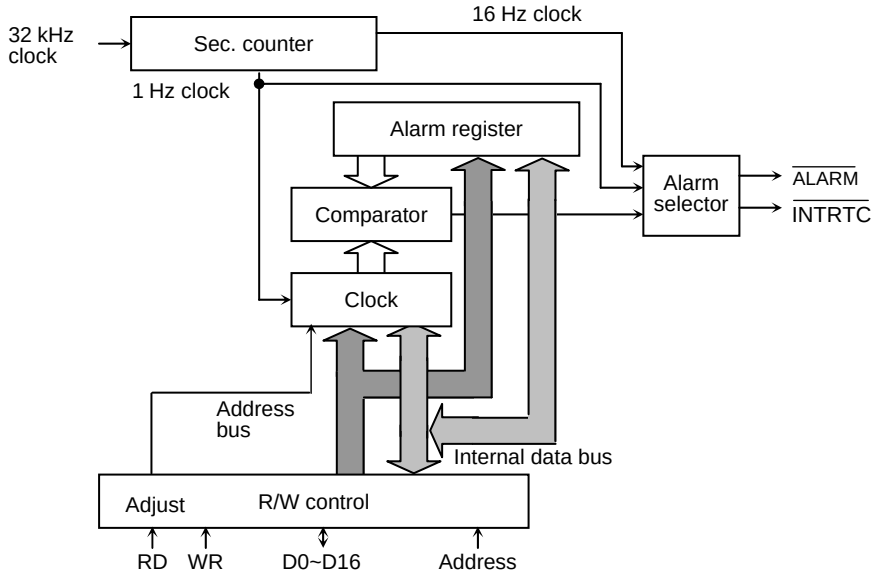


Fig. 18-1 Block Diagram

(Note 1) Western calendar year column:
This product uses only the final two digits of the year. The year following 99 is 00 years. Please take into account the first two digits when handling years in the western calendar.

(Note 2) Leap year:
A leap year is divisible by 4 excluding a year divisible by 100; the year divisible by 100 is not considered to be a leap year. Any year divisible by 400 is a leap year. This product is considered the year divisible by 4 to be a leap year and does not take into account the above exceptions. It needs adjustments for the exceptions.

18.3 Control Registers

Reset operation initializes the following registers:

- RTCPAGER<PAGE>,<ADJUST>,<INTENA>
- RTCRESTR<RSTALM>,<RSTTMR>,<DIS16HZ>,<DIS1HZ>

Other clock-related registers are not initialized by reset operation.

Before starting the RTC, set the time, month, day, day of the week, year and leap year in the relevant registers.

Caution is required in setting clock data, adjusting seconds or resetting the clock. Refer to “18.5.3 Entering the Low Power Consumption Mode”.

Table 18-1 PAGE0 (clock function) register

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
RTCSECR	0x4004_0100		40 sec	20 sec	10 sec	8 sec	4 sec	2 sec	1 sec	Second column	R/W
RTCMINR	0x4004_0101		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
RTCHOURR	0x4004_0102			20 hours /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column	R/W
RTCDAYR	0x4004_0104						W2	W1	W0	Day of the week column	R/W
RTCDATER	0x4004_0105			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
RTCMONTHR	0x4004_0106				Oct.	Aug.	Apr.	Feb.	Jan.	Month column	R/W
RTCYEARR	0x4004_0107	Year 80	Year 40	Year 20	Year 10	Year 8	Year 4	Year 2	Year 1	Year column (lower two columns)	R/W
RTCPAGER	0x4004_0108	Interrupt enable			Adjustment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RTCRESTR	0x4004_010C	1Hz enable	16Hz enable	Clock reset	Alarm reset	Always write “0”.				Reset register	W only

(Note) Reading RTC**SECR,** RTC**MINR,** RTC**HOURR,** RTC**DAYR,** RTC**MONTHR,** RTC**YEARR of PAGE0 captures the current state.**

Table 18-2 PAGE1 (alarm function) registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
RTCSECR	0x4004_0100										
RTCMINR	0x4004_0101		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
RTCHOURR	0x4004_0102			20 hours /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column	R/W
RTCDAYR	0x4004_0104						W2	W1	W0	Day of the week column	R/W
RTCDATER	0x4004_0105			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
RTCMONTHR	0x4004_0106								24/12	24-hours clock mode	R/W
RTCYEARR	0x4004_0107							Leap-year setting		Leap-year mode	R/W
RTCPAGER	0x4004_0108	Interrupt enable			Adjustment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W,R/W
RTCRESTR	0x4004_010C	1Hz enable	16Hz enable	Clock reset	Alarm reset	Always write “0”.				Reset register	W only

(Note 1) Reading RTC**SECR,** RTC**MINR,** RTC**HOURR,** RTC**DAYR,** RTC**MONTHR,** RTC**YEARR of PAGE1 captures the current state.**

(Note 2) RTC**SECR,** RTC**MINR,** RTC**HOURR,** RTC**DAYR,** RTC**MONTHR,** RTC**YEARR of PAGE0 and YEARR of PAGE1 (for leap year) must be read twice and compare the data captured.**

18.4 Detailed Description of Control Register

The RTC is not initialized by system reset. All registers must be initialized at the beginning of the program.

(1) Second column register (for PAGE0 only)

RTCSECR

	7	6	5	4	3	2	1	0
Bit symbol	—	SE6	SE5	SE4	SE3	SE2	SE1	SE0
Read/Write	R	R/W						
After reset	0	Undefined						
Function	"0" is read.	40 sec. column	20 sec. column	10 sec. column	8 sec. column	4 sec. column	2 sec. column	1 sec. column

0	0	0	0	0	0	0	0 sec
0	0	0	0	0	0	1	1 sec
0	0	0	0	0	1	0	2 sec
0	0	0	0	0	1	1	3 sec
0	0	0	0	1	0	0	4 sec
0	0	0	0	1	0	1	5 sec
0	0	0	0	1	1	0	6 sec
0	0	0	0	1	1	1	7 sec
0	0	0	1	0	0	0	8 sec
0	0	0	1	0	0	1	9 sec
0	0	1	0	0	0	0	10 sec
			:				
0	0	1	1	0	0	1	19 sec
0	1	0	0	0	0	0	20 sec
			:				
0	1	0	1	0	0	1	29 sec
0	1	1	0	0	0	0	30 sec
			:				
0	1	1	1	0	0	1	39 sec
1	0	0	0	0	0	0	40 sec
			:				
1	0	0	1	0	0	1	49 sec
1	0	1	0	0	0	0	50 sec
			:				
1	0	1	1	0	0	1	59 sec

Note) The setting other than listed above is prohibited.

(2) Minute column register (for PAGE0/1)

RTCMINR

	7	6	5	4	3	2	1	0
Bit symbol	—	MI6	MI5	MI4	MI3	MI2	MI1	MI0
Read/Write	R	R/W						
After reset	0	Undefined						
Function	"0" is read	40 min. column	20 min. column	10 min. column	8 min. column	4 min. column	2 min. column	1 min. column

0	0	0	0	0	0	0	0 min
0	0	0	0	0	0	1	1 min
0	0	0	0	0	1	0	2 min
0	0	0	0	0	1	1	3 min
0	0	0	0	1	0	0	4 min
0	0	0	0	1	0	1	5 min
0	0	0	0	1	1	0	6 min
0	0	0	0	1	1	1	7 min
0	0	0	1	0	0	0	8 min
0	0	0	1	0	0	1	9 min
0	0	1	0	0	0	0	10 min
			:				
0	0	1	1	0	0	1	19 min
0	1	0	0	0	0	0	20 min
			:				
0	1	0	1	0	0	1	29 min
0	1	1	0	0	0	0	30 min
			:				
0	1	1	1	0	0	1	39 min
1	0	0	0	0	0	0	40 min
			:				
1	0	0	1	0	0	1	49 min
1	0	1	0	0	0	0	50 min
			:				
1	0	1	1	0	0	1	59 min

Note) The setting other than listed above is prohibited.

(3) Hour column register (for PAGE0/1)

1. 24-hour clock mode (RTCMONTHR<MO0>="1")

RTCHOURR

	7	6	5	4	3	2	1	0
Bit symbol	—		HO5	HO4	HO3	HO2	HO1	HO0
Read/Write	R		R/W					
After reset	0		Undefined					
Function	"0"is read.		20 hours column	10 hours column	8 hours column	4 hours column	2 hours column	1 hour column

0	0	0	0	0	0	0 o' clock
0	0	0	0	0	1	1 o' clock
0	0	0	0	1	0	2 o' clock
			:			
0	0	1	0	0	0	8 o' clock
0	0	1	0	0	1	9 o' clock
0	1	0	0	0	0	10 o' clock
			:			
0	1	1	0	0	1	19 o' clock
1	0	0	0	0	0	20 o' clock
			:			
1	0	0	0	1	1	23 o' clock

Note) The setting other than listed above is prohibited.

2. 12-hour clock mode (RTCMONTHR<MO0>="0")

RTCHOURR

	7	6	5	4	3	2	1	0
Bit symbol	—		HO5	HO4	HO3	HO2	HO1	HO0
Read/Write	R		R/W					
After reset	0		Undefined					
Function	"0"is read.		PM/AM	10 hours column	8 hours column	4 hours column	2 hours column	1 hour column

0	0	0	0	0	0	0 o' clock (AM)
0	0	0	0	0	1	1 o' clock
0	0	0	0	1	0	2 o' clock
			:			
0	0	1	0	0	1	9 o' clock
0	1	0	0	0	0	10 o' clock
0	1	0	0	0	1	11 o' clock
1	0	0	0	0	0	0 o' clock (PM)
1	0	0	0	0	1	1 o' clock

Note) The setting other than listed above is prohibited.

(4) Day of the week column register (for PAGE0/1)

RTCDAYR

	7	6	5	4	3	2	1	0
Bit symbol	—					WE2	WE1	WE0
Read/Write	R					R/W		
After reset	0					Undefined		
Function	"0" is read.					W2	W1	W0

0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

Note) The setting other than listed above is prohibited.

(5) Day column register (PAGE0/1)

RTCDATER

	7	6	5	4	3	2	1	0
Bit symbol	—		DA5	DA4	DA3	DA2	DA1	DA0
Read/Write	R		R/W					
After reset	0		Undefined					
Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

0	0	0	0	0	0	0
0	0	0	0	0	1	1st day
0	0	0	0	1	0	2nd day
0	0	0	0	1	1	3rd day
0	0	0	1	0	0	4th day
			:			
0	0	1	0	0	1	9th day
0	1	0	0	0	0	10th day
0	1	0	0	0	1	11th day
			:			
0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day
			:			
1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

Note 1) The setting other than listed above is prohibited.

Note 2) Do not set for non-existent days (e.g.: 30th Feb.)

(6) Month column register (for PAGE0 only)

RTCMONTHR

	7	6	5	4	3	2	1	0
Bit symbol	—			MO4	MO4	MO2	MO1	MO0
Read/Write	R			R/W				
After reset	0			Undefined				
Function	“0” is read.			10 months	8 months	4 months	2 months	1 month

0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May
0	0	1	1	0	June
0	0	1	1	1	July
0	1	0	0	0	August
0	1	0	0	1	September
1	0	0	0	0	October
1	0	0	0	1	November
1	0	0	1	0	December

Note) The setting other than listed above is prohibited.

(7) Selection of 24-hour clock or 12-hour clock (for PAGE1 only)

RTCMONTHR

	7	6	5	4	3	2	1	0
Bit symbol	—							MO0
Read/Write	R							R/W
After reset	0							Undefined
Function	“0” is read.							1: 24-hours 0: 12-hours

（Note）Do not change the RTC**MONTHR<MO0> bit while the RTC is in operation.**

(8) Year column register (for PAGE0 only)

RTCYEARR

	7	6	5	4	3	2	1	0
Bit symbol	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
Read/Write	R/W							
After reset	Undefined							
Function	80 years	40 years	20 years	10 years	8 years	4 years	2 years	1 year

0	0	0	0	0	0	0	0	00 year
0	0	0	0	0	0	0	1	01 year
0	0	0	0	0	0	1	0	02 years
0	0	0	0	0	0	1	1	03 years
0	0	0	0	0	1	0	0	04 years
0	0	0	0	0	1	0	1	05 years
				:				
1	0	0	1	1	0	0	1	99 years

Note) The setting other than listed above is prohibited.

(9) Leap year register (for PAGE1 only)

RTCYEARR

	7	6	5	4	3	2	1	0
Bit symbol	—						LEAP1	LEAP0
Read/Write	R						R/W	
After reset	0						Undefined	
Function	“0” is read.						00: leap year 01: one year after leap year 10: two years after leap year 11: three years after leap year	

0	0	Current year is a leap-year.
0	1	Current year is the year following a leap-year.
1	0	Current year is two years after a leap year.
1	1	Current year is three years after a leap year

(10) PAGE register (for PAGE0/1)

		7	6	5	4	3	2	1	0
RTCPAGER	Bit symbol	INTENA	—		ADJUST	ENATMR	ENAALM	—	PAGE
	Read/Write	R/W	R		R/W	R/W		R	R/W
	After reset	0	0		0	Undefined		0	0
A read-modify-write operation cannot be performed.	Function	INTRTC 0: Disabled 1: Enabled	"0" is read.		[Write] 0: Don't care 1:Sets ADJUST request [Read] 0:No ADJUST requested 1:ADJUST requested	Clock 0: Disabled 1: Enabled	ALARM 0: Disabled 1: Enabled	"0" is read.	PAGE selection

(Note) Keep the setting order of <ENATMR>, <ENAAML> and <INTENA> as shown in the example below. Ensure an interval of time between Clock/Alarm and interrupt.

Example: Clock setting/Alarm setting

	7	6	5	4	3	2	1	0	
RTCPAGER ←	0	0	0	0	1	1	0	0	Enables Clock and alarm
RTCPAGER ←	1	0	0	0	1	1	0	0	Enables interrupt by setting bit 7 to "1".

PAGE	0	Selects Page0
	1	Selects Page1

ADJUST	0	Don't care
	1	Adjusts seconds. The request is sampled when the sec. counter counts up. If the time elapsed is between 0 and 29 seconds, the sec. counter is cleared to "0". If the time elapsed is between 30 and 59 seconds, the min. counter is carried and sec. counter is cleared to "0". Reading this bit shows if ADJUST is requested or not.

(11) Reset register (for PAGE0/1)

RTCRESTR

A read-modify-write operation cannot be performed.

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	–	DIS2HZ	DIS4HZ	DIS8HZ
Read/Write	R/W				R	R/W		
After reset	1	1	0	0	0	1	1	1
Function	1 Hz 0: Enabled 1: Disabled	16 Hz 0: Enabled 1: Disabled	[Write] 0:Don't care 1:Clock reset [Read] 0:No RESET request-ed 1:RESET request-ed	0:Don't care 1:Alarm reset	"0" is read.	2 Hz 0: Enabled 1: Disabled	4 Hz 0: Enabled 1: Disabled	8 Hz 0: Enabled 1: Disabled

RSTALM	0	Unused
	1	Initializes alarm registers (Minute Column, Hour Column, Day Column and Day of the week Column) as follows. Minute: 00, Hour: 00, Day: 01, Day of the week: Sunday

RSTTMR	0	Unused
	1	Resets sec counter. Reading this bit shows if RESET is requested or not. The request is sampled using low-speed clock.

<DIS1HZ>	<DIS2HZ>	<DIS4HZ>	<DIS8HZ>	<DIS16HZ>	RTCPAGER <ENAALM>	Interrupt source signal
1	1	1	1	1	1	Alarm
0	1	1	1	1	0	1Hz
1	0	1	1	1	0	2Hz
1	1	0	1	1	0	4Hz
1	1	1	0	1	0	8Hz
1	1	1	1	0	0	16Hz
Others						Outputs "0".

18.5 Operational Description

The RTC incorporates a sec. counter that generates a 1Hz signal from a 32.768 KHz signal. The sec. counter operation must be taken into account when using the RTC.

Note : After reset, low-speed oscillator stops oscillation and XT1/XT2 pins are initialized to port-P (PP0,PP1). Please re-setup to RTC registers.

18.5.1 Reading clock data

1. Using 1Hz interrupt

   The 1Hz interrupt is generated being synchronized with counting up of the sec. counter. Data can be read correctly if reading data after 1Hz interrupt occurred.

2. Using pair reading

   There is a possibility that the clock data may be read incorrectly if the internal counter operates carry during reading. To ensure correct data reading, read the clock data twice as shown below. A pair of data read successively needs to match.

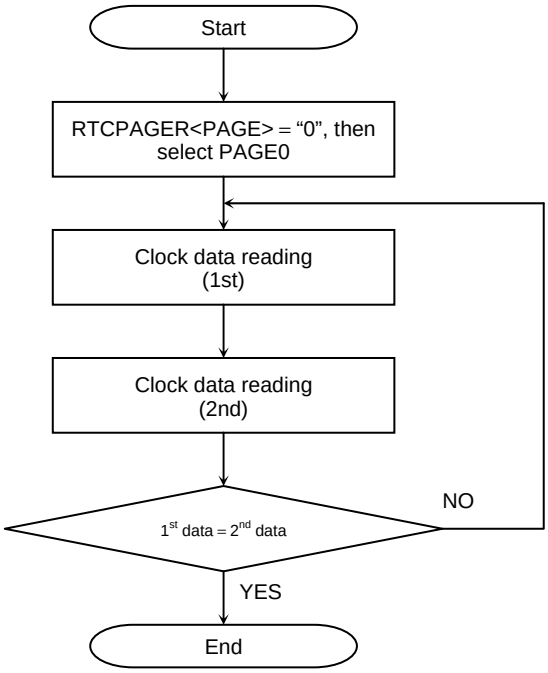


Fig. 18-2 Flowchart of the clock data reading

18.5.2 Writing clock data

A carry during writing ruins correct data writing. The following procedure ensures the correct data writing.

1. Using 1Hz interrupt

The 1Hz interrupt is generated being synchronized with counting up of the sec. counter. If data is written in the time between 1Hz interrupt and subsequent one second count, it completes correctly.

2. Resetting counter

Write data after resetting the sec. counter.

The 1Hz-interrupt is generated one second after enabling the interrupt subsequent to counter reset. The time must be set within one second after the interrupt.

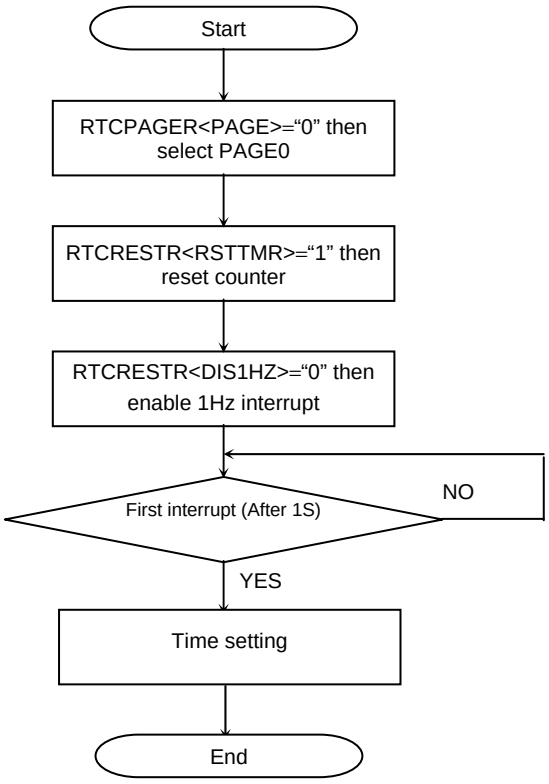


Fig. 18-3 Flowchart of the clock data writing

3. Disabling the clock

Writing “0” to RTCPAGER<ENATMR> disables clock operation including a carry.

Stop the clock after the 1Hz-interrupt. The sec. counter keeps counting. Set the clock again and enable the clock within one second before next 1Hz-interrupt.

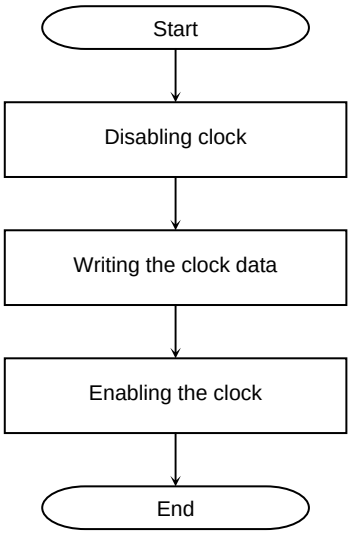


Fig. 18-4 Flowchart of the disabling clock

18.5.3 Entering the Low Power Consumption Mode

To enter SLEEP mode, in which the system clock stops, after changing clock data, adjusting seconds or resetting the clock, be sure to observe one of the following procedures:

1. After changing the clock setting registers, setting the RTCPAGER<ADJUST> bit or setting the RTCRESTR<RSTTMR> bit, wait for one second for an interrupt to be generated.
2. After changing the clock setting registers, setting the RTCPAGER<ADJUST> bit or setting the RTCRESTR<RSTTMR> bit, read the corresponding clock register values, <ADJUST> or <RSTTMR> to make sure that the setting you have made is reflected.

18.6 Alarm Function

By writing “1” to RTCPAGER<PAGE>, the alarm function of the PAGE1 registers is enabled. One of the following three signals is output to the $\overline{\text{ALARM}}$ pin.

(1) “0” pulse (when the alarm register corresponds with the clock)

(2) 1Hz cycle “0” pulse

(3) 16Hz cycle “0” pulse

In any cases shown above, the INTRTC outputs one cycle pulse of low-speed clock. It outputs the INTRTC interrupt request simultaneously.

The INTRTC interrupt signal is falling edge triggered. Specify the falling edge as the active state in the CG Interrupt Mode Control Register.

(1) “0” pulse (when the alarm register corresponds with the clock)

“0” pulse is output to the $\overline{\text{ALARM}}$ pin when the values of the PAGE0 clock register and the PAGE1 alarm register correspond. The INTRTC interrupt is generated and the alarm is triggered.

The alarm settings

Initialize the alarm with alarm prohibited. Write “1” to RTCRESTR<RSTALM>. It makes the alarm setting to be 00 minute, 00 hour, 01 day and Sunday.

Setting alarm for min., hour, date and day is done by writing data to the relevant PAGE1 register. Enable the alarm with the RTCPAGER <ENAALM> bit. Enable the interrupt with the RTCPAGER <INTENA> bit.

The following is an example program for outputting an alarm from the $\overline{\text{ALARM}}$ pin at noon (12:00p.m.) on Monday 5th.

	7	6	5	4	3	2	1	0	
RTCPAGER ←	0	0	0	0	1	0	0	1	Disables alarm, sets PAGE1
RTCRESTR ←	1	1	0	1	0	0	0	0	Initializes alarm
RTCDAYR ←	0	0	0	0	0	0	0	1	Monday
RTCDATER ←	0	0	0	0	0	1	0	1	5th day
RTCHOURR ←	0	0	0	1	0	0	1	0	Sets 12 o’clock
RTCMINR ←	0	0	0	0	0	0	0	0	Sets 00 min.
RTCPAGER ←	0	0	0	0	1	1	0	0	Enables alarm
RTCPAGER ←	1	0	0	0	1	1	0	0	Enables interrupt

The above alarm works in synchronization with the low-speed clock. When the CPU is operating at high frequency oscillation, a maximum of one clock delay at 32 kHz (about 30μs) may occur for the time register setting to become valid.

(Note) To make the alarm work repeatedly (e.g. every Wednesday at 12:00), next alarm must be set during the INTRTC interrupt routine that is generated when the time set for the alarm matches the RTC count.

(2) 1Hz cycle “0” pulse

The RTC outputs a "0" pulse cycle of low-speed 1Hz clock to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>=1 after setting RTCPAGER<ENAALM>= “0”, RTCRESTR<DIS1HZ>= “0” and <DIS16HZ>= “1”. It generates an INTRTC interrupt simultaneously.

(3) 16Hz cycle “0” pulse

The RTC outputs a "0" pulse cycle of low-speed 16Hz clock to the $\overline{\text{ALARM}}$ pin by setting RTCPAGER<INTENA>=1 after setting RTCPAGER<ENAALM>= “0”, RTCRESTR<DIS1HZ>= “1” and <DIS16HZ>= “0”. It generates an INTRTC interrupt simultaneously.

19 Ocsillation Frequency Detector (OFD)

19.1 Brief overview

The oscillation frequency detector circuit generates a reset for micro if the external oscillation of high frequency for CPU clock exceeds the detection frequency range.(Note)

The following is the operation flow how to use OFD.

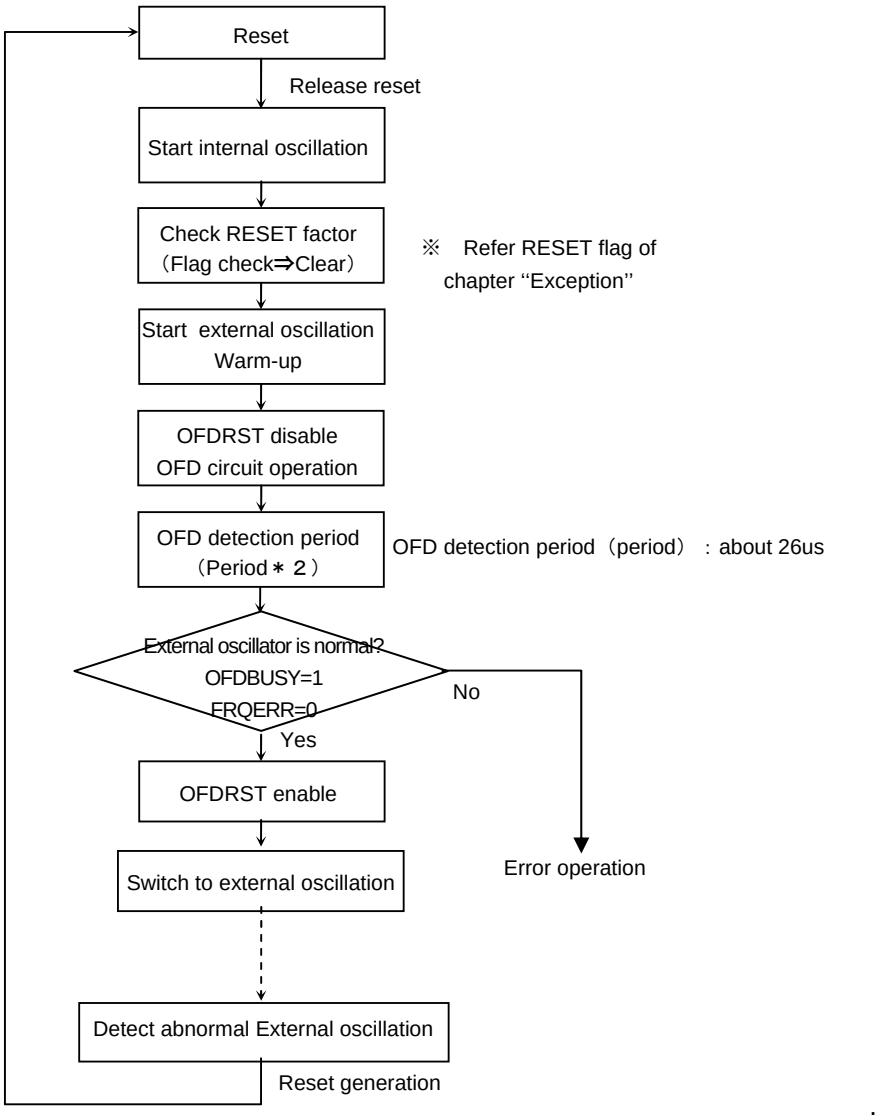


Figure 19-1 Oscillation Frequency Detector Circuit Operation Flow

TMPM380 can be operated by both internal OSC and external OSC, however OFD circuit can detect only external OSC clock frequency using internal OSC clock.

Note) OFD circuit is designed for detecting abnormal oscillation however it is not guaranteed that OFD can detect all defects at any time. Therefore please design the system carefully assuming there is some possibility that OFD circuit will not work correctly.

19.2 Configuration

The oscillation frequency detection circuit is controlled by OFDCR1, OFDCR2 registers and the detection frequency range is specified by OFDMX, OFDMN which are the detection frequency setting registers. The lower detection frequency is specified by OFDMN registers and the higher detection frequency is specified by OFDMX registers. Figure 19-2 shows the example of frequency detection range.

When the oscillation frequency detection is enabled, writing to OFDCR2, OFDMX, OFDMN, OFDRST registers is disabled. Therefore, the setting the detection frequency to these registers should be done when the oscillation frequency detection is disabled. And writing to OFDCR2, OFDMX, OFDMN, OFDRST registers is controlled by OFDCR1 register. To write OFDCR2, OFDMX, OFDMN, OFDRST registers, the write enable code "0xF9" should be set to OFDCR1 beforehand. To enable the oscillation frequency detector, set "0xE4" to OFDCR2 after setting "0xF9" to OFDCR1. Since the oscillation frequency detection is disabled after an external reset input or POR (Power On Reset), write "0xF9" to OFDCR1 and write "0xE4" to OFDCR2 register to enable its function.

When the TMPM380 detects the the frequency which is out of the range setting by OFDMX and OFDMN registers, OFD(Oscillatiion Frequency Detection) reset will be generated and starting operation with internal oscillator . (OFD circuit is disable) By the OFD reset, all I/Os except power supply pins, RESET, X1,X2 and debug pins (PB3-PB7) are initialized as high impedance. If OFD reset is generated by detecting the stopping of high frequency, the internal circuitries such as registers hold the condition at the timing of oscillation stop. To initialize these internal circuitries, an external re-starting of oscillation is needed.

All registers of oscillation frequency detector (OFDCR1, OFDCR2, OFDMX, OFDMN, OFDRST, OFDSTAT) are initialized by the reset generated from oscillation frequency detector, and then, TMPM380 re-starts from internal OSC without detection of external OSC‘s frequency by OFD. Therefore it is recommended to check which reset factor was occurred by the flag register in intial routine. For the details, please refer the RSTFLG: Reset Flag Register

Note) The oscillation frequency detection reset is available only in NORMAL and IDLE modes. Before shifting to STOP mode, SLOW mode and SLEEP mode, disable the oscillation frequency detection by software

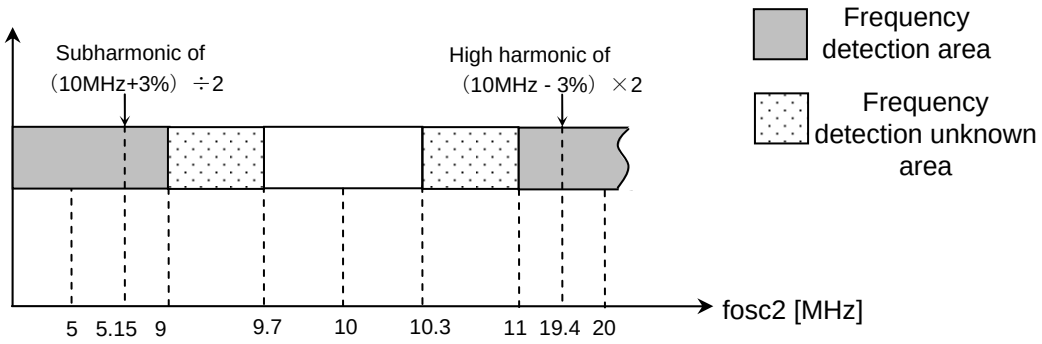


Figure 19-2 Detect frequency range (Example: 10MHz)

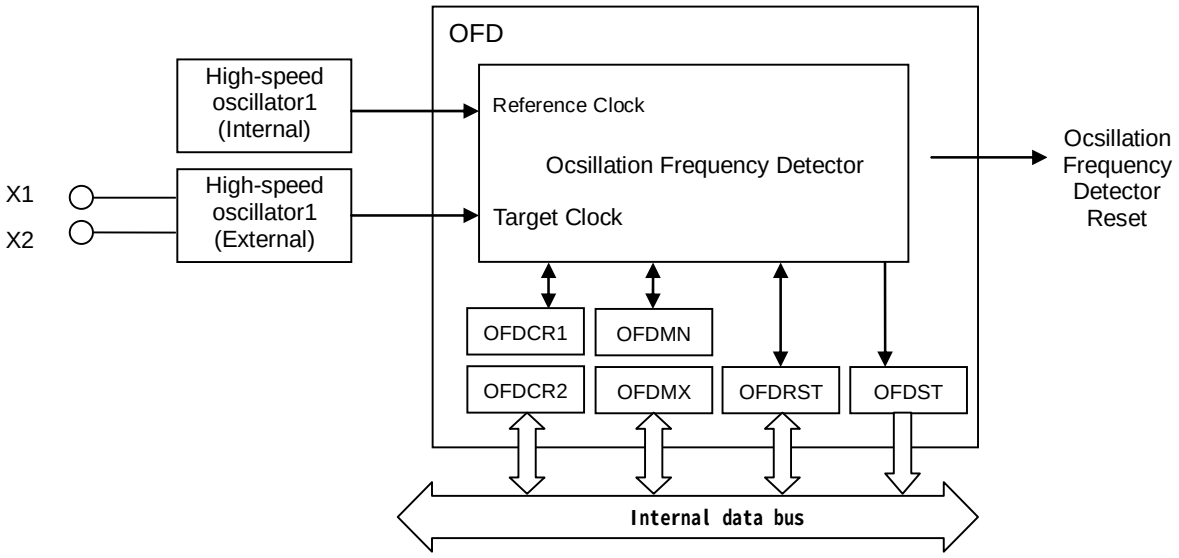


Figure 19-3 Internal Circut connection of OFD (Outline)

19.3 Control

The oscillation frequency detection is controlled by oscillation frequency detection control register 2 (OFDCR2). The detection frequency is specified by lower/higher detection frequency setting registers (OFDMN, and OFDMX). Writing to OFDCR2, OFDMN, OFDMX, OFDRST is controlled by oscillation frequency detection control register 1 (OFDCR1).

19.3.1.1 Oscillation frequency detection control register1(OFDCR1)

OFDCR1: 0x4004_0800

	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit symbol	OFDWEN7	OFDWEN6	OFDWEN5	OFDWEN4	OFDWEN3	OFDWEN2	OFDWEN1	OFDWEN0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	1	1	0
Function	0x06: Disabling of writing to OFDCR2/OFDMN/OFDMX (Write disable code) 0xF9: Enabling of writing to OFDCR2/OFDMN/OFDMX(Write enable code) Others: Reserved (Note 1)							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							

Note1) Only "0x06" and "0xF9" is valid to OFDCR1. If other value than "0x06" and "0xF9" is written to OFDCR1, "0x06" is written to OFDCR1 automatically.

19.3.1.2 Oscillation frequency detection control register2 (OFDCR2)

OFDCR2: 0x4004_0804

	7	6	5	4	3	2	1	0
Bit symbol	OFDEN7	OFDEN6	OFDEN5	OFDEN4	OFDEN3	OFDEN2	OFDEN1	OFDEN0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	0x00: Disabling of oscillation frequency detection 0xE4: Enabling of oscillation frequency detection Others: Reserved (Note 1)							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							

Note1) Only "0x00" and "0xE4" is valid to OFDCR2. Writing other value than "0x00" and "0xE4" to OFDCR2 is ignored.

Norw2) Writing to OFDCR2 is protected by setting "0x06" to OFDCR1 but reading from OFDCR2 is always enabled without setting of OFDCR1.

19.3.1.3 Lower detection frequency setting register (OFDMN)

OFDMN: 0x4004_0808

	7	6	5	4	3	2	1	0
Bit symbol	OFDMN7	OFDMN6	OFDMN5	OFDMN4	OFDMN3	OFDMN2	OFDMN1	OFDMN0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Set the count value of lower detection frequency							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							

19.3.1.4 Higher detection frequency setting register (OFDMX)

OFDMX: 0x4004_0810

	7	6	5	4	3	2	1	0
Bit symbol	OFDMX7	OFDMX6	OFDMX5	OFDMX4	OFDMX3	OFDMX2	OFDMX1	OFDMX0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Set the count value of higher detection frequency							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							

Note1) OFDMN and OFDMX can not be written when the oscillation frequency detection circuit is enabled (OFDCR2="0xE4") or writing is disabled with OFDCR1="0x06". An attempt to write OFDMN and OFDMX can not complete a write operation.

Note2) Writing to OFDMN and OFDMX is protected by setting "0x06" to OFDCR1 but reading from OFDMN and OFDMX is always enabled without setting of OFDCR1.

19.3.1.5 Oscillation frequency detector reset enable control register (OFDRST)

OFDRST: 0x4004_0818

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	-	OFDRSTEN
Read/Write	R	R	R	R	R	R	R	R/W
After reset	0	0	0	0	0	0	0	1
Function	Always read “0”							1: Enable 0: Disable
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							

Note1) Writing to OFDRST is protected by setting "0x06" to OFDCR1 but reading from OFDRST is always enabled without setting of OFDCR1.

19.3.1.6 Oscillation frequency detector Status register (OFDSTAT)

OFDSTAT: 0x4004_081C

	7	6	5	4	3	2	1	0
Bit symbol	-	-	-	-	-	-	OFDBUSY	FRQERR
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”						OFD Operation 1: Run 2: Stop	Frequency Status flag 1: Error 0: No Error
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit symbol	-	-	-	-	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Always read “0”							

19.4 Function

19.4.1 Enabling and Disabling the Oscillation Frequency Detection

Writing "0xE4" to OFDCR2 with OFDCR1="0xF9" enables the oscillation frequency detection, and writing "0x00" to OFDCR2 with OFDCR1="0xF9" disables the oscillation frequency detection.

Setting "0xF9" to OFDCR1 enables writing to OFDCR2 and setting "0x06" to OFDCR1 disables writing to OFDCR2. Reading from OFDCR2 is always enabled without setting of OFDCR1. OFDCR1 is initialized to "0x06" by external reset and OFDCR2 is initialized to "0x00" by external reset.

After writing data to OFDCR2, set "0x06" to OFDCR1 to protect OFDCR2 register.

The oscillation frequency detection is available only in NORMAL and IDLE mode. Table 19-1 shows the availability of oscillation frequency detector.

Table 19-1 Availability of oscillation frequency detector

<table>
<tr><th>Operation Mode</th><th>Operation of OFD circuit
(OFDCR2=0xE4)</th><th>All I/Os condition after reset by OFD
(Except power supply, RESET pins)</th></tr>
<tr><td>NORAML</td><td>Available</td><td>High impedance</td></tr>
<tr><td>IDLE</td><td>Available</td><td>High impedance</td></tr>
<tr><td>SLOW</td><td colspan="2" rowspan="3">OFD circuit observes high frequency ocsillation only.
Be disabled by software before entering this mode</td></tr>
<tr><td>SLEEP</td></tr>
<tr><td>STOP
(including warnning up period)</td></tr>
<tr><td>RESET by oscillation frequency detection reset</td><td>Be initialize (Disable)</td><td>High impedance</td></tr>
<tr><td>RESET by internal reset (Note1)</td><td>Be initialize (Disable)</td><td>High impedance</td></tr>
<tr><td>RESET by external reset</td><td>Be initialize (Disable)</td><td>-</td></tr>
</table>

Note1) Internal reset: Watchdog timer reset, OFD reset, SYSRESETREQ reset

TMPM380 has 4-kind of system reset source (POR, external reset input, WDT, OFD). And TMPM380 has 2-kind of high clock OSC (external, internal). Following timming diagram shows the operation and timing of various reset factor including OFD and internal and external clock.

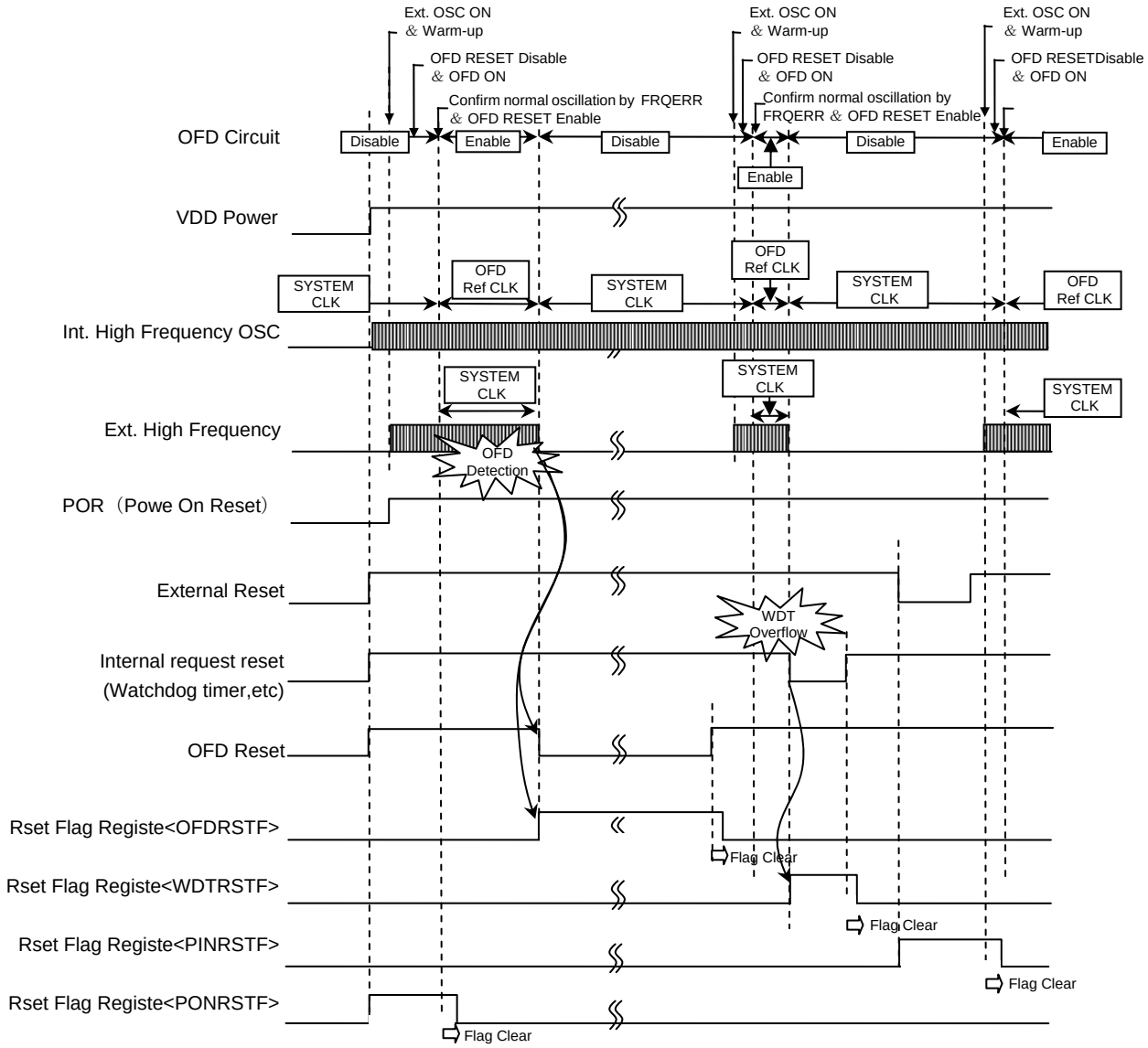


Figure 19-4 The relation of OFD operation and various RESETs

19.4.2 Setting the Lower and Higher Frequency for Detection

The detection frequency is specified by OFDMN and OFDMX registers. The relation between the setting value to these registers and the detection frequency is shown in Table 19-2.

Table 19-2 High frequency and Setting value for detection frequency（Tentativ）

OFDMN	Detection frequency range [MHz]	Non detection frequency range [MHz]	OFDMX	Detection frequency range [MHz]	Non detection frequency range [MHz]
45	5.6	7.0	64	9.9	8.1
46	5.8	7.2	65	10.1	8.2
47	5.9	7.3	66	10.3	8.3
48	6.0	7.5	67	10.4	8.4
49	6.2	7.6	68	10.6	8.6
50	6.3	7.8	69	10.7	8.7
51	6.4	7.9	70	10.9	8.8
52	6.5	8.1	71	11.0	8.9
53	6.7	8.2	72	11.2	9.1
54	6.8	8.4	73	11.3	9.2
55	6.9	8.6	74	11.5	9.3
56	7.0	8.7	75	11.7	9.4
57	7.2	8.9	76	11.8	9.6
58	7.3	9.0	77	12.0	9.7
59	7.4	9.2	78	12.1	9.8
60	7.5	9.3	79	12.3	9.9
61	7.7	9.5	80	12.4	10.1
62	7.8	9.6	81	12.6	10.2
63	7.9	9.8	82	12.7	10.3
64	8.1	9.9	83	12.9	10.5
			84	13.0	10.6
			85	13.2	10.7
			86	13.4	10.8
			87	13.5	11.0

19.4.3 Oscillation Frequency Detection Reset

If the TMPM380 detects lower frequency specified by OFDMN or higher frequency specified by OFDMX, the oscillation frequency detector outputs a reset signal for all I/Os.

And then, TMPM380 re-start from internal OSC without detection of external OSC‘s frequency by OFD.

RESET genaration of OFD is controlled by OFDRST<OFDRSTEN>. When OFD reset function is enabled:OFDRST<OFDRSTEN>=1 after OFDCR1 is written for enable code:“F9H”, reset by OFD is issued.

When OFD reset function is disabled:OFDRST<OFDRSTEN>=0, reset by OFD is not issued. Even OFD reset function is disabled:OFDRST<OFDRSTEN>=0, status flag:OFDSTAT<FRQERR> can be checked.

20. Power-on Reset Circuit (POR)

The power-on reset circuit generates a reset when the power is turned on. When the supply voltage is lower than the detection voltage of the power-on reset circuit, a power-on reset signal is generated.

20.1 Configuration

The power-on reset circuit consists of a reference voltage generation circuit, a comparator and a power-on counter.

The supply voltage divided by ladder resistor is compared with the voltage generated by the reference voltage generation circuit by the comparator.

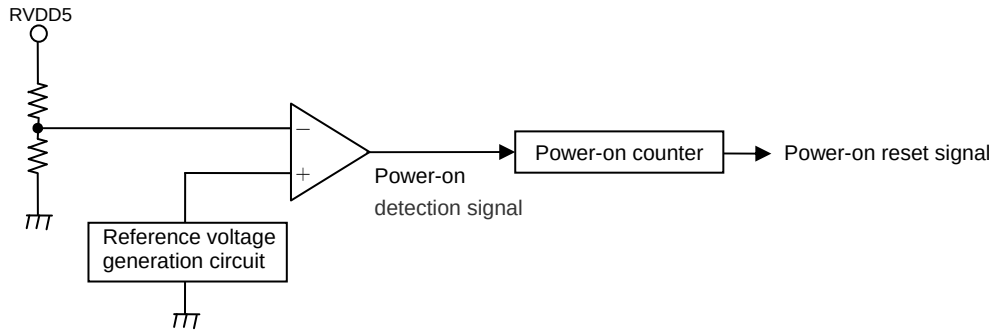


Figure 20-1 Power-on Reset Circuit

20.2 Function

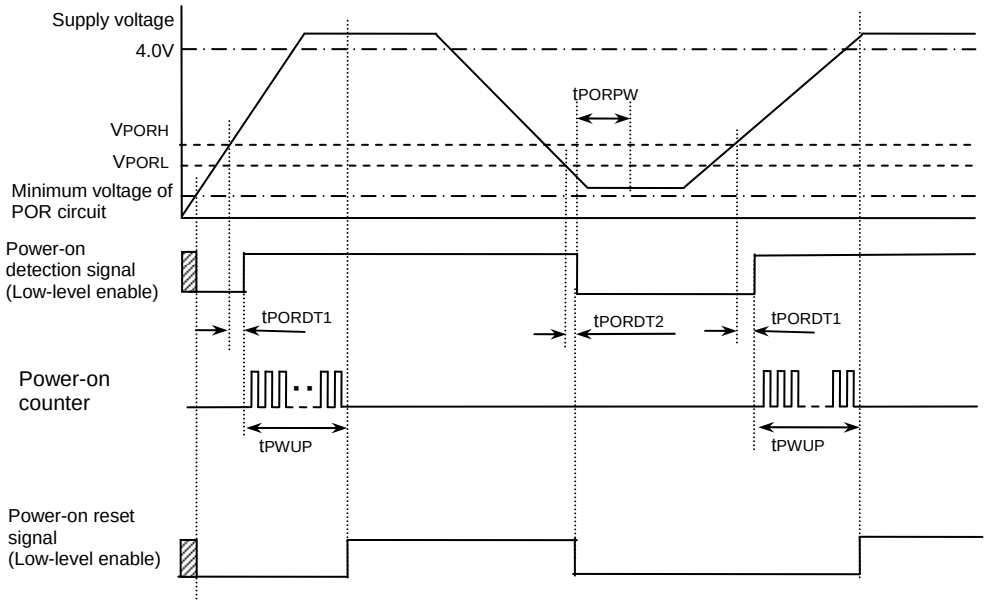
When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated. If the power supply voltage exceeds the releasing voltage of the power-on reset circuit, power-on counter is activated and $2^{13}/f_{osc}$ (s) later, a power-on reset signal is released.

When power supply voltage goes down, if the supply voltage is equal to or lower than the detecting voltage of the power-on reset circuit, a power-on reset signal is generated.

During the generation of power-on reset, the power-on counter circuit, the CPU and peripheral circuits are reset.

When the power-on reset cuicuit is activated without an external reset input signal, the supply voltage should be increased to the recommended operating voltage range (Note) within 0.6ms from the detection of the releasing voltage of the power-on reset circuit. If the supply voltage does not reach the range, the TMPM380/M382 cannot operate properly.

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



Note 1: The power-on reset circuit may operate improperly, depending on fluctuations in the supply voltage. Refer to the electrical characteristics and take them into consideration when designing equipment.

Note 2: If the supply voltage is lower than the minimum voltage of Power-on Reset circuit in which the circuit cannot operate properly, the power-on reset signal becomes undefined value.

Figure 20-2 Operation Timing of Power-on Reset

Symbol	Parameter	Min	Typ.	Max	Unit
V_{PORH}	Power-on Reset releasing voltage	2.8	3	3.2	V
V_{PORL}	Power-on Reset detection voltage	2.6	2.8	3.0	V
t_{PORDT1}	Power-on Reset release response time		30		μs
t_{PORDT2}	Power-on Reset detection response time		30		μs
t_{PORPW}	Power-on Reset minimum pulse width	45			μs
t_{PWUP}	Power-on counter (Note 2)		$2^{13}/f_{osc}$		s

Note 1 : Since the power-on reset releasing voltage and the power-on reset detection voltage relatively change, the detection voltage is never reversed.

Note 2 : 3.2ms at 10MHz.

For the details about Power-on sequence, refer to the chapter of “Electrical Characteristics”.

For the details about how to use external reset input, refer to “reset exceptions” in the chapter of “Exceptions”.

21. Voltage Detection Circuit (VLTD)

The voltage detection circuit detects any decrease in the supply voltage and generates NMI.

Note: The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (RVDD5). Refer to the electrical characteristics and take them into consideration when designing equipment.

21.1 Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (RVDD5) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDLVL), and the comparator compares it with the reference voltage.

When the supply voltage (RVDD5) becomes lower than the detection voltage (VDLVL), a voltage detection interrupt (NMI) is generated.

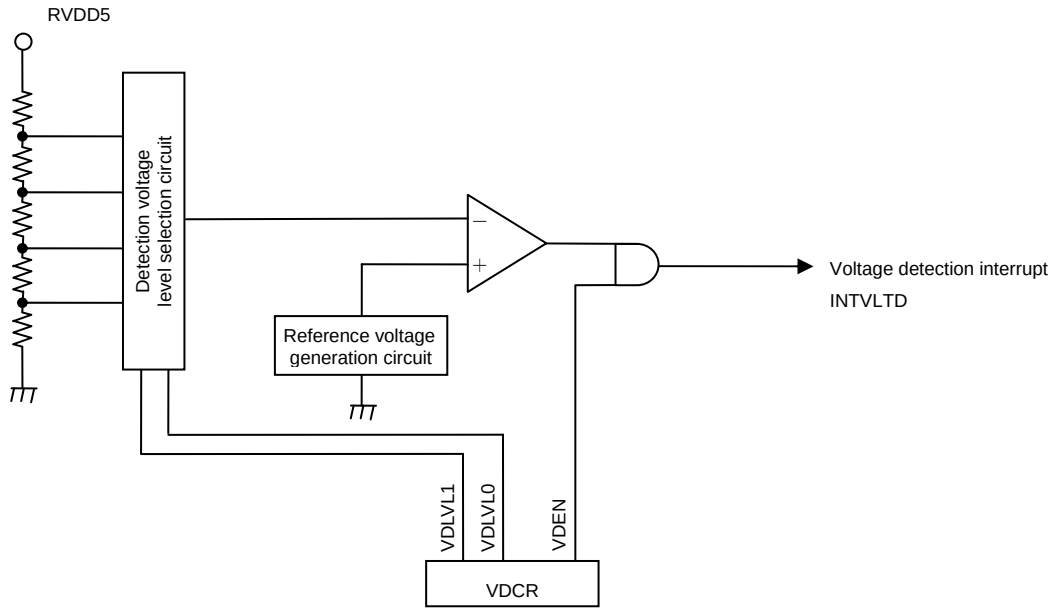


Figure 21-1 Voltage Detection Circuit

21.2 Control

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

Voltage detection control register

VDCR (0x4004_0900)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	VDLVL1	VDLVL0	VDEN
Read/Write	R	R	R	R	R	R/W		R/W
After reset	0	0	0	0	0	00		0

VDLVL[1:0]	Selection for detection voltage	00 : 3.8 ± 0.2 V 01 : 4.1 ± 0.2 V 10 : 4.4 ± 0.2 V 11 : 4.6 ± 0.2 V
VDEN	Enables/disables the operation of voltage detection	0 : Disables the operation of voltage detection 1 : Enables the operation of voltage detection

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

Voltage detection status register

VDSR (0x4004_0904)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	VDSR
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

VDSR	Voltage detection status register	0 : Power supply voltage is higher than the detection voltage specified by VDLVL[1:0]. 1 : Power supply voltage is lower than the detection voltage specified by VDLVL[1:0].

21.3 Function

The detection voltage can be selected by VDCR<VDLVL[1:0]>. Enabling/disabling the voltage detection can be programmed by VDCR<VDEN>.

After the voltage detection operation is enabled, When the supply voltage (RVDD5) becomes lower than the detection voltage <VDLVL[1:0]>, a voltage detection interrupt (NMI) is generated.

21.3.1 Enabling/disabling the voltage detection operation

Setting VDCR<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

VDCR<VDEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (RVDD5) is lower than the detection voltage (VDLVL), setting VDCR<VDEN> to "1" generates a voltage detection interrupt (NMI) at the time.

21.3.2 Selecting the detection voltage level

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

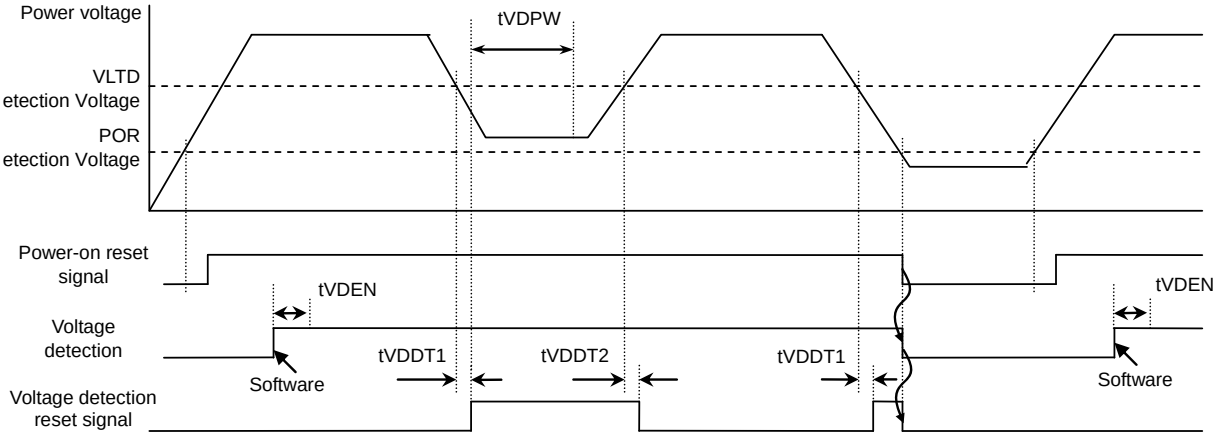


Figure 21-2 Voltage Detection Timing

Symbol	Parameter	Min	Typ.	Max	Unit
tVDEN	Setup time after enabling voltage detection		40		μs
tVDDT1	Voltage detection response time		40		μs
tVDDT2	Voltage detection releasing time		40		μs
tVDPW	Voltage detection minimum pulse width	45			μs

22 DMA Controller (DMAC)

TMPM380/M382 has a DMA controller controlled by DMA request select registers

22.1 Function Overview

The table below lists its major functions.

Table 22-1 DMA controller functions

Item	Function		Overview
Number of channels	2ch		
	Hardware start		Supports 14 types of DMA requests for peripheral IPs.
	Software start		Started with a write to the DMACSoftBReq register.
Bus master	32bit×1 (AHB)		
Priority	DMA channel 0 (high) to DMA channel 1 (low)		Fixed by hardware
FIFO	4word × 2ch		
Bus width	8/16/32bit		Settable individually for transfer source and destination
Burst size	1/4/8/16/32/64/128/256		
Number of transfers	up to 4095		
Address	Transfer source address	incr / no-incr	It is possible to specify whether Source and Destination addresses should increment or should not increment (should be fixed). (Address wrapping is not supported.)
	Transfer destination address	incr / no-incr	
Endian	Only little endian is supported.		
Transfer type	Peripheral circuit (register) → memory Memory → peripheral circuit (register) Memory → memory (Note1)		When "memory → memory" is selected, hardware start for DMA startup is not supported. See the DMACCxConfiguration register for more information.
Interrupt function	Transfer end interrupt Error interrupt		
Special Function	Scatter/gather function		

* 1 word = 32 bits

Note1) Follows transfer type is not supported
From Peripheral circuit (register) to Peripheral circuit (register)

22.2 DMA transfer type

Transfer type are supported following 3-kinds type.
The condition of each transfer type are showed the following table.

Table 22-2 DMA transfer type

	DMA direction	DMA request circuit	Support DMA request (Note2)	Other condtion
1	Memory → peripheral circuit (register)	peripheral circuit (Destination)	Burst request	In case of 1word transmission, set to the “1” for burst size of DMA controller.
2	Peripheral circuit (register) → memory	peripheral circuit（Source）	Burst request /single request (Note1)	If the amount of data transfer is not an integral multiple of the burst size, both burst and single transfers are used. Amount of remaining transfer data ≥ Burst size • Uses burst transfer. Amount of remaining transfer data < Burst size • Uses single transfer
3	Memory → memory	DMAC	-	Start condition: Enabling the DMAC starts data transfer with no DMAC request required. Use condition: Transfer of all transfer data is complete. The DMAC channel is disabled.

Note1) Single request circuit in this Micro controller: SSP
Note2) Internal connection of DMA request: Refer the next page please.

22.3 Block Diagram

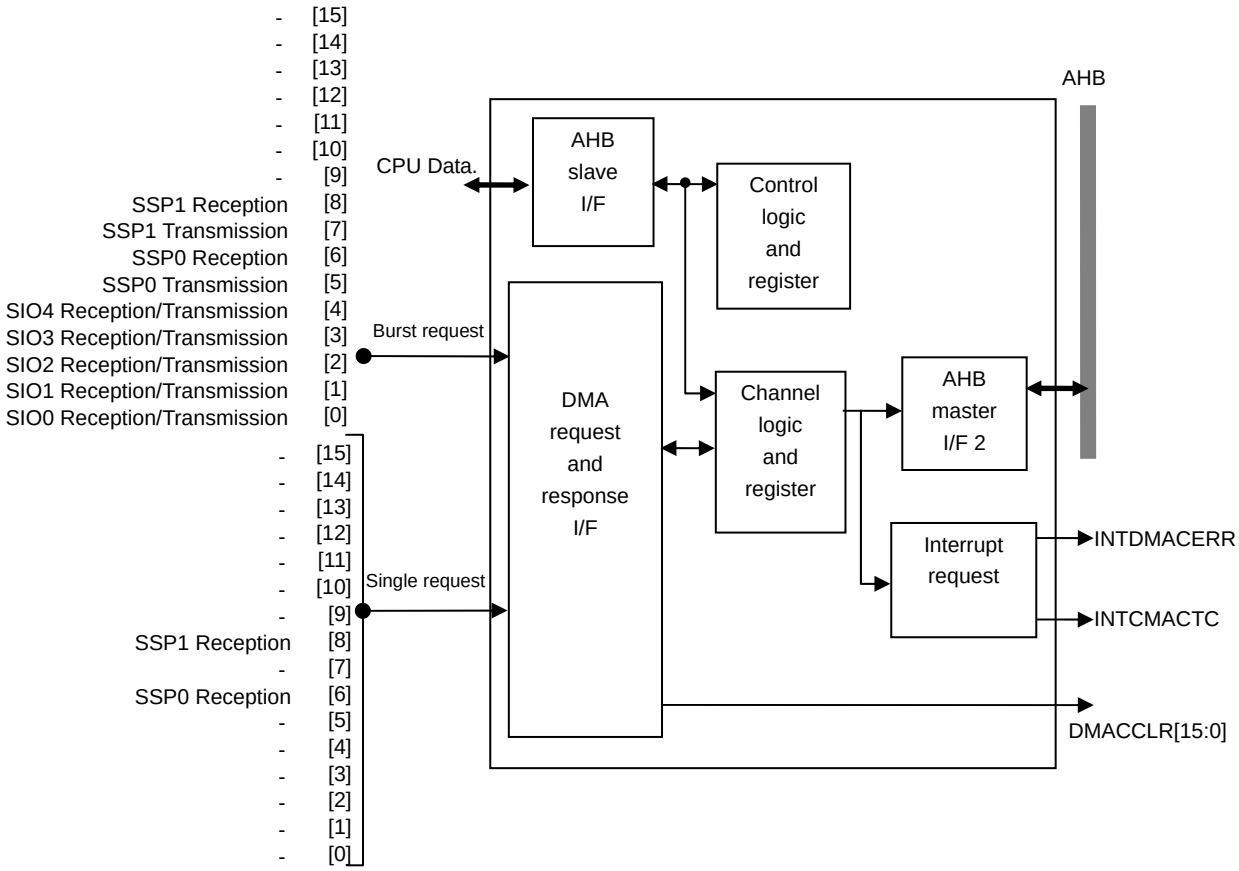


Figure 22-1 Block diagram

Table 22-3 DMA request number chart

DMA request No.	Corresponding peripheral	
	Burst	Single
0	SIO0 Reception / Transmission	-
1	SIO1 Reception / Transmission	-
2	SIO2 Reception / Transmission	-
3	SIO3 Reception / Transmission	-
4	SIO4 Reception / Transmission	-
5	SSP0 Transmission	-
6	SSP0 Reception	SSP0 Reception
7	SSP1 Transmission	-
8	SSP1 Reception	SSP1 Reception
9	-	-
10	-	-
11	-	-
12	-	-
13	-	-
14	-	-
15	-	-

22.4 Description of Registers

22.4.1 DMAC register list

The following lists the SFRs and their functions:

Table22-4 SFR list

Base address= 0x4008_0000

Register Name	Address (base+)	Description
DMACIntStaus	0x0000	DMAC Interrupt Status Register
DMACIntTCStatus	0x0004	DMAC Interrupt Terminal Count Status Register
DMACIntTCClear	0x0008	DMAC Interrupt Terminal Count Clear Register
DMACIntErrorStatus	0x000C	DMAC Interrupt Error Status Register
DMACIntErrClr	0x0010	DMAC Interrupt Error Clear Register
DMACRawIntTCStatus	0x0014	DMAC Raw Interrupt Terminal Count Status Register
DMACRawIntErrorStatus	0x0018	DMAC Raw Error Interrupt Status Register
DMACEnbldChns	0x001C	DMAC Enabled Channel Register
DMACSoftBReq	0x0020	DMAC Software Burst Request Register
DMACSoftSReq	0x0024	DMAC Software Single Request Register
–	0x0028	Reserved
–	0x002C	Reserved
DMACConfiguration	0x0030	DMAC Configuration Register
–	0x0034	Reserved
DMACC0SrcAddr	0x0100	DMAC Channel0 Source Address Register
DMACC0DestAddr	0x0104	DMAC Channel0 Destination Address Register
DMACC0LLI	0x0108	DMAC Channel0 Linked List Item Register
DMACC0Control	0x010C	DMAC Channel0 Control Register
DMACC0Configuration	0x0110	DMAC Channel0 Configuration Register
DMACC1SrcAddr	0x0120	DMAC Channel1 Source Address Register
DMACC1DestAddr	0x0124	DMAC Channel1 Destination Address Register
DMACC1LLI	0x0128	DMAC Channel1 Linked List Item Register
DMACC1Control	0x012C	DMAC Channel1 Control Register
DMACC1Configuration	0x0130	DMAC Channel1 Configuration Register

Note) Access the registers by using word reads and word writes.

22.4.2 DMACIntStatus (DMAC Interrupt Status Register)

Address = (0x4008_0000) + 0x0000

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntStatus1	R	0y0	Status of DMAC channel 1 interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested
[0]	IntStatus0	R	0y0	Status of DMAC channel 0 interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested

[Explanation]

a. <IntStatus[1:0]>

Status of the DMAC interrupt generation after passing through the transfer end interrupt enable register and error interrupt enable register. An interrupt is requested when there is a transfer error or when the counter completes counting.

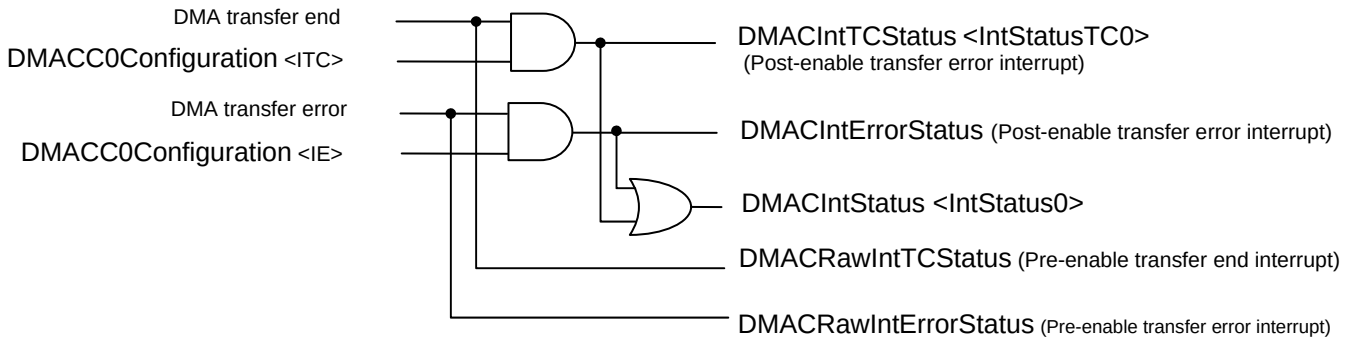


Figure 22-2 Interrupt-related block diagram

22.4.3 DMACIntTCStatus (DMAC Interrupt Terminal Count Status Register)

Address = (0x4008_0000) + 0x0004

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntTCStatus1	R	0y0	Status of DMAC channel 1 transfer end interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested
[0]	IntTCStatusT0	R	0y0	Status of DMAC channel 0 transfer end interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested

[Explanation]

b. <IntTCStatus[1:0]>

The status of post-enable transfer end interrupt generation.

22.4.4 DMACIntTCClear (DMAC Interrupt Terminal Count Clear Register)

Address = (0x4008_0000) + 0x0008

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntTCClear1	W	0y0	Clear DMAC channel 1 transfer end interrupt 0y1: Clear 0y0: Do nothing
[0]	IntTCClear0	W	0y0	Clear DMAC channel 0 transfer end interrupt 0y1: Clear 0y0: Do nothing

[Explanation]

c. <IntTCClearCH[1:0]>

The DMACINTTCS register bit of the channel that corresponds to the bit to which "1" was written will clear the interrupt.

22.4.5 DMACIntErrorStatus (DMAC Interrupt Error Status Register)

Address = (0x4008_0000) + 0x000C

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntErrStatus1	R	0y0	Status of DMAC channel 1 error interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested
[0]	IntErrStatus0	R	0y0	Status of DMAC channel 0 error interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested

[Explanation]

d. <IntErrStatus[1:0]>
Error interrupt status after enabled

22.4.6 DMACIntErrClr (DMAC Interrupt Error Clear Register)

Address = (0x4008_0000) + 0x0010

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	IntErrClr1	W	0y0	Clear DMAC channel 1 error interrupt 0y1: Clear 0y0: Do nothing
[0]	IntErrClr0	W	0y0	Clear DMAC channel 0 error interrupt 0y1: Clear 0y0: Do nothing

[Explanation]

e. <IntErrClr[1:0]>

"1": Clears an error interrupt request.

22.4.7 DMACRawIntTCStatus (DMAC Raw Interrupt Terminal Count Status Register)

Address = (0x4008_0000) + 0x0014

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RawIntTCS1	R	0y0	Status of DMAC channel 1 pre-enable transfer end interrupt generation 0y1: Interrupt requested 0y0: Interrupt not requested
[0]	RawIntTCS0	R	0y0	Status of DMAC channel 0 pre-enable transfer end interrupt generation 0y1: Interrupt requested 0y0: Interrupt not requested

[Explanation]

f. < RawIntTCS [1:0]>

"1": Interrupt requested for Pre-enable transfer end interrupt.

22.4.8 DMACRawIntErrorStatus (DMAC Raw Error Interrupt Status Register)

Address = (0x4008_0000) + 0x0018

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	RawIntErrS1	R	0y0	Status of DMAC channel 1 pre-enable error interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested
[0]	RawIntErrS0	R	0y0	Status of DMAC channel 0 pre-enable error interrupt generation. 0y1: Interrupt requested 0y0: Interrupt not requested

[Explanation]

g. < RawIntErrS [1:0]>

"1": Interrupt requested for Pre-enable error interrupt.

22.4.9 DMACEnbldChns (DMAC Enabled Channel Register)

Address = (0x4008_0000) + 0x001C

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	EnabledCH1	R	0y0	DMA channel 1 enable status 0y1: Enable 0y0: Disable
[0]	EnabledCH0	R	0y0	DMA channel 0 enable status 0y1: Enable 0y0: Disable

[Explanation]

h. <EnabledCH[1:0]>

"0": The bits of the appropriate channel are cleared when DMA transfer is complete.

"1": The appropriate channel DMA is enabled.

22.4.10 DMACSoftBReq (DMAC Software Burst Request Register)

Address = (0x4008_0000) + 0x0020

Bit	Bit Symbol	Type	Reset Value	Description
[31:9]	–	–	Undefined	Read undefined. Write as zero.
[8]	SoftBReq8	R/W	0y0	DMA burst request by software at. SSP1 reception 0y1: DMA burst requested 0y0: Disabled (WR)
[7]	SoftBReq7	R/W	0y0	DMA burst request by software at SSP1 transmission 0y1: DMA burst requested 0y0: Disabled (WR)
[6]	SoftBReq6	R/W	0y0	DMA burst request by software at SSP0 reception 0y1: DMA burst requested 0y0: Disabled (WR)
[5]	SoftBReq5	R/W	0y0	DMA burst request by software at SSP0 transmission 0y1: DMA burst requested 0y0: Disabled (WR)
[4]	SoftBReq4	R/W	0y0	DMA burst request by software at SIO4 transmission and reception 0y1: DMA burst requested 0y0: Disabled (WR)
[3]	SoftBReq3	R/W	0y0	DMA burst request by software at SIO3 transmission and reception 0y1: DMA burst requested 0y0: Disabled (WR)
[2]	SoftBReq2	R/W	0y0	DMA burst request by software at SIO2 transmission and reception 0y1: DMA burst requested 0y0: Disabled (WR)
[1]	SoftBReq1	R/W	0y0	DMA burst request by software at SIO1 transmission and reception 0y1: DMA burst requested 0y0: Disabled (WR)
[0]	SoftBReq0	R/W	0y0	DMA burst request by software at SIO0 transmission and reception 0y1: DMA burst requested 0y0: Disabled (WR)

[Explanation]

i. <SoftBReq[8:0]>

Sets a DMA burst transfer request by software. When the DMA burst transfer by software is complete, the appropriate bits in SoftBReq[8:0] are cleared.

Note) Do not execute DMA requests by software and hardware peripheral at the same time.

22.4.11 DMACSoftSReq (DMAC Software Single Request Register)

Address = (0x4008_0000) + 0x0024

Bit	Bit Symbol	Type	Reset Value	Description
[31:9]	–	–	Undefined	Read undefined. Write as zero.
[8]	SoftSReq8	R/W	0y0	DMA single request by software at SSP1 reception 0y1: Generate a DMA single request 0y0: Disabled (WR)
[7]	–	–	Undefined	Read undefined. Write as zero.
[6]	SoftSReq6	R/W	0y0	DMA single request by software at SSP0 reception 0y1: Generate a DMA single request 0y0: Disabled (WR)
[5:0]	–	–	Undefined	Read undefined. Write as zero.

[Explanation]

j. <SoftSReq[8]>, <SoftSReq[6]>
Sets a DMA single transfer request by software. When the DMA single transfer by software is complete, SoftSReq[8] or SoftSReq[6] are cleared.

Note) Do not execute a DMA request by software when a DMA request by hardware peripheral is generated.

22.4.12 DMACConfiguration (DMAC Configuration Register)

Address = (0x4008_0000) + 0x0030

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	–	–	Undefined	Read undefined. Write as zero.
[1]	M	R/W	0y0	DMA1 endian types: 0: Little endian 1: Reserved
[0]	E	R/W	0y0	DMA circuit control: 0 : Stop 1 : Operate

[Explanation]

k. <M>

DMA endian configuration

l. <E>

The registers for the DMA circuit cannot be written or read unless the DMA circuit operates. When operating the DMA, always keep the DMA circuit operating.

22.4.13 DMACC0SrcAddr (DMAC Channel0 Source Address Register)

Address = (0x4008_0000) + 0x0100

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	SrcAddr	R/W	0x00000000	Sets a DMA transfer source address

[Explanation]

m. <SrcAddr>

Because enabling channels updates the data written in the registers, set DMACCxSrcAddr before enabling the channels.

When the DMA is operating, the value in the DMACCxSrcAddr register sequentially changes, so the read values are not fixed.

Do not update DMACC0SrcAddr during transfer. To change the value, be sure to set the DMACCxConfiguration register to disable the channel before change.

- DMACCxSrcAddr (DMAC Channel x Source Address Register) (x = 1)

Refer to the description on DMACC0SrcAddr because the structures and explanations on the above registers are the same as DMACC0SrcAddr. Also, refer to Table22-4 SFR list for register names and addresses.

22.4.14 DMACC0DestAddr (DMAC Channel0 Destination Address Register)

Address = (0x4008_0000) + 0x0104

Bit	Bit Symbol	Type	Reset Value	Description
[31:0]	DestAddr	R/W	0x00000000	Sets a DMA transfer destination address

[Explanation]

n. <DestAddr>

Do not update DMACC0DestAddr during transfer. To change the value, be sure to set the DMACCxConfiguration register to disable the channel before change.

- DMACCxDestAddr (DMAC Channel x Destination Address Register) (x = 1)

  Refer to the description on DMACC0DestAddr because the structures and explanations on the above registers are the same as DMACC0DestAddr. Also, refer to Table22-4 SFR list for register names and addresses.

22.4.15 DMACC0LLI (DMAC Channel0 Linked List Item Register)

Address = (0x4008_0000) + 0x0108

Bit	Bit Symbol	Type	Reset Value	Description
[31:2]	LLI	R/W	0x00000000	Sets the first address of the next transfer information.
[1:0]	–	–	Undefined	Read undefined. Write as zero.

[Explanation]

o. <LLI>

Set a value smaller than 0xFFFF_FFF0 for <LLI>.

When <LLI> = 0, currently, LLI is the last chain. After DMA transfer finishes, the DMA channel is disabled.

* For detailed operation, see 22.5 “Special function”

- DMACCxLLI (DMAC Channel x Linked List Item Register) (x = 1)

  Refer to the description on DMACC0LLI because the structures and explanations on the above registers are the same as DMACC0LLI. Also, refer to Table22-4 SFR list for register names and addresses.

22.4.16 DMACC0Control (DMAC Channel0 Control Register)

Address = (0x4008_0000) + 0x010C

Bit	Bit Symbol	Type	Reset Value	Description
[31]	I	R/W	0y0	Register for enabling a transfer end interrupt when the scatter/gather function is used 0y0 : Disable 0y1 : Enable
[30:28]	–	–	Undefined	Read undefined. Write as zero.
[27]	DI	R/W	0y0	Increment the transfer destination address 0y0: Do not increment 0y1: Increment
[26]	SI	R/W	0y0	Increment the transfer source address 0y0: Do not increment 0y1: Increment
[25:24]	–	–	Undefined	Read undefined. Write as zero.
[23:21]	Dwidth[2:0]	R/W	0y000	Transfer destination bit width 0y000: Byte (8 bits) 0y001: Half-word (16 bits) 0y010: Word (32 bits) other: Reserved
[20:18]	Swidth[2:0]	R/W	0y000	Transfer source bit width 0y000: Byte (8 bits) 0y001: Half-word (16 bits) 0y010: Word (32 bits) other: Reserved
[17:15]	DBSize[2:0]	R/W	0y000	Transfer destination burst size: 0y000: 1 beat 0y001: 4 beats 0y010: 8 beats 0y011: 16 beats 0y100: 32 beats 0y101: 64 beats 0y110: 128 beats 0y111: 256 beats
[14:12]	SBSize[2:0]	R/W	0y000	Transfer source burst size: 0y000: 1 beat 0y001: 4 beats 0y010: 8 beats 0y011: 16 beats 0y100: 32 beats 0y101: 64 beats 0y110: 128 beats 0y111: 256 beats
[11:0]	TransferSize	R/W	0x000	Set the total number of transfers

[Explanation] The same explanation is applied for the other channels too.

p. <I>

Register for enabling a transfer end interrupt when the scatter/gather function is used

q. <DI>

Increments the address of a transfer destination.

Depends on the bit width of the transfer source. Increments the address, each depending on Dwidth as follows:

8-bit : 1 byte

16-bit : 2 bytes

32-bit : 4 bytes

r. <SI>

Increments the address of a transfer destination.

Depends on the bit width of the transfer source. Increments the address, each depending on Swidth as follows:

8-bit : 1 byte

16-bit : 2 bytes

32-bit : 4 bytes

s. <Swidth[2:0]>

Set a transfer destination bit width so that the transfer size becomes an integral multiple of the transfer destination bit width.

t. <DBSize[2:0]>

Note) The burst size to be set with DBsize has nothing to do with the HBURST for the AHB bus.

u. <SBSize[2:0]>

Note) The burst size to be set with SBsize has nothing to do with the HBURST for the AHB bus.

v. <TransferSize>

Set the total number of transfers when the DMAC is used as the flow controller.

This value decrements to "0" as DMAC transfer is executed. The read operation reads the number of transfers that have not been executed yet.

The total number of transfers is used as the unit for the transfer source bit width.

ex: When Swidth=8bit, the number of transfers is expressed in the units of byte.

ex: When Swidth=16bit, the number of transfers is expressed in the units of half word.

ex: When Swidth=32bit, the number of transfers is expressed in the units of word.

Note) If the transfer source bit width is smaller than the transfer destination bit width, care must be taken when setting

the total number of transfers.

Set the number so that the following expression is satisfied:

Transfer source bit width × total number of transfers = Transfer destination bit width × N

N: Integer number

- DMACCxControl (DMAC Channel x Control Register) (x = 1)

  Refer to the description on DMACC0Control because the structures and explanations on the above registers are the same as DMACC0Control. Also, refer to Table22-4 SFR list for register names and addresses.

22.4.17 DMACC0Configuration (DMAC Channel0 Configuration Register)

Address = (0x4008_0000) + 0x0110

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>Type</th><th>Reset Value</th><th>Description</th></tr>
<tr><td>[31:19]</td><td>–</td><td>–</td><td>Undefined</td><td>Read undefined. Write as zero.</td></tr>
<tr><td>[18]</td><td>Halt</td><td>R/W</td><td>0y0</td><td>0y0: Accept a DMA request
0y1: Ignore a DMA request</td></tr>
<tr><td>[17]</td><td>Active</td><td>R</td><td>0y0</td><td>0y0: No data exists in the FIFO
0y1: Data exists in the FIFO</td></tr>
<tr><td>[16]</td><td>Lock</td><td>R/W</td><td>0y0</td><td>0y0: Disable locked transfer
0y1: Enable locked transfer</td></tr>
<tr><td>[15]</td><td>ITC</td><td>R/W</td><td>0y0</td><td>Transfer end interrupt enable register
0y0: Disable interrupt
0y1: Enable interrupt</td></tr>
<tr><td>[14]</td><td>IE</td><td>R/W</td><td>0y0</td><td>Error interrupt enable register
0y0: Disable interrupt
0y1: Enable interrupt</td></tr>
<tr><td>[13:11]</td><td>FlowCntrl</td><td>R/W</td><td>0y000</td><td>
<table>
<tr><th>FlowCntrl setting value</th><th>Transfer method</th></tr>
<tr><td>0y000</td><td>Memory to Memory</td></tr>
<tr><td>0y001</td><td>Memory to Peripheral</td></tr>
<tr><td>0y010</td><td>Peripheral to Memory</td></tr>
<tr><td>0y011</td><td>Reserved</td></tr>
</table>
0y100~0y111: Reserved
</td></tr>
<tr><td>[10]</td><td>–</td><td>–</td><td>Undefined</td><td>Read undefined. Write as zero.</td></tr>
<tr><td>[9:6]</td><td>DestPeripheral</td><td>R/W</td><td>0y000</td><td>Transfer destination peripheral (Note 1)
0y000~0y1111</td></tr>
<tr><td>[5]</td><td>–</td><td>–</td><td>Undefined</td><td>Read undefined. Write as zero.</td></tr>
<tr><td>[4:1]</td><td>SrcPeripheral</td><td>R/W</td><td>0y000</td><td>Transfer source peripheral (Note 1)
0y000~0y1111</td></tr>
<tr><td>[0]</td><td>E</td><td>R/W</td><td>0y0</td><td>Channel enable
0y0 : Disable
0y1 : Enable</td></tr>
</table>

Note) Refer to DMA request number chart.

[Explanation]

w. <Halt>

Halts DMA.

x. <Active>

Indicates whether data is present in the channel FIFO.

y. <Lock>

Sets a locked transfer (Non-divided transfer). When locked transfer is enabled, as many burst transfers as specified are consecutively executed without releasing the bus.

z. <ITC>

Transfer end interrupt enable

aa. <IE>

Error interrupt enable

bb. <FlowCntrl>

Sets a transfer method.

0y000: Memory to Memory

0y001: Memory to Peripheral

0y010: Peripheral to Memory

0y011: Reserved

0y100~0y111: Reserved

Note) When "memory to memory" is selected, hardware start for DMA startup is not supported. Writing to <E>= 1 starts transfer.

cc. <DestPeripheral>

The DMA request peripheral number is expressed by binary.

When a memory is the transfer destination, this setting is ignored.

dd. <SrcPeripheral>

The DMA request peripheral number is expressed by binary.

When a memory is the transfer source, this setting is ignored.

ee. <E>

This bit can be used to enable/disable the channels. Disabling channels during transfer loses the data in the FIFO. Initialize all the channels before restart.

To pause the transfer, stop the DMA request by using the <HALT> bit, and poll the data until the <Active> bit becomes "0" and then disable the channel with the <E> bit.

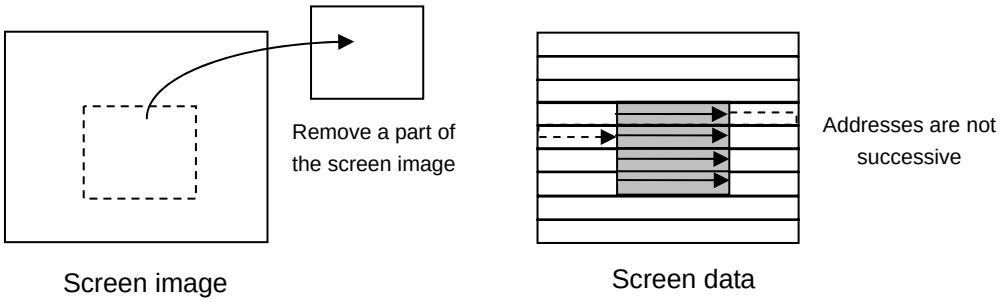
- DMACCxConfiguration (DMAC Channel x Configuration Register) (x = 1)

  Refer to the description on DMACC0Configuration because the structures and explanations on the above registers are the same as DMACC0Configuration. Also, refer to Table22-4 SFR list for register names and addresses.

22.5 Special Functions

22.5.1 Scatter/gather function

When removing a part of image data and transferring it, image data cannot be handled as consecutive data, and the address changes dramatically depending on the special rule. Since DMA can transfer data only by using consecutive addresses, it is necessary to make required settings at locations where addresses changes.



The scatter/gather function can consecutively operate DMA settings (transfer source address, destination address, number of transfers, and transfer bus width) by re-loading them each time a specified number of DMA executions have completed via a pre-set "Linked List" where the CPU does not need to control the operation.

Setting "1" in the DMACCxLLI register enables/disables the operation.

The items that can be set with Linked List are configured with the following 4 words:

1) DMACCxSrcAddr
2) DMACCxDestAddr
3) DMACCxLLI
4) DMACCxControl

They can be used with the interrupt operation.

An interrupt depends on the Terminal Count Interrupt enable bit of the DMACCxControl register, and can be generated at the end of each LLI. When this bit is used, a condition can be added even during transfer using LLI to perform branch operation, etc. To clear the interrupt, control the appropriate bit of the DMACIntTCClear register.

22.5.2 Linked list operation

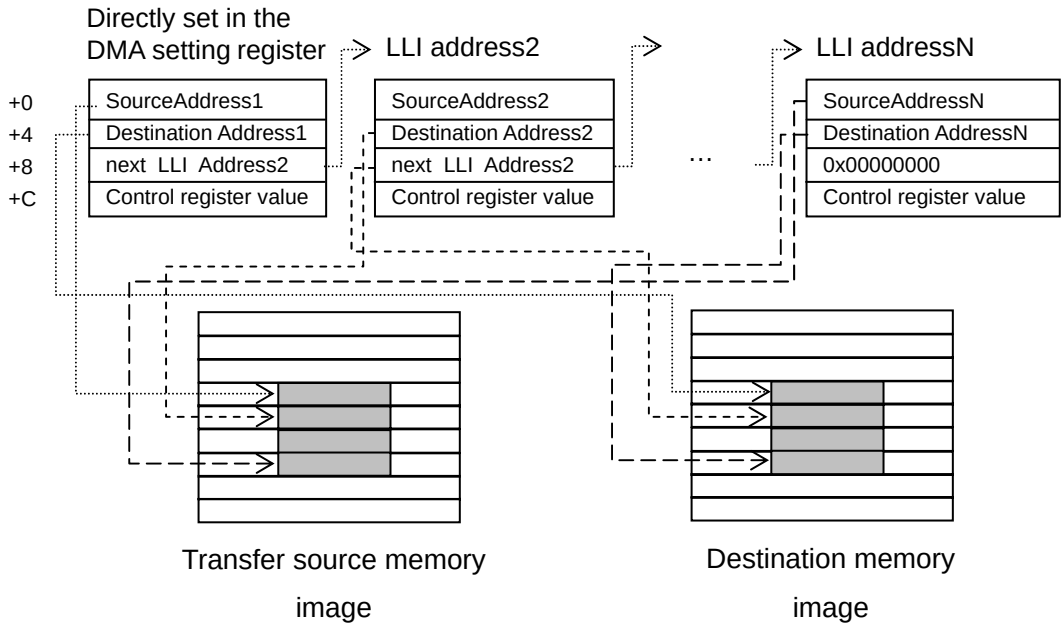
To operate the scatter/gather function, a transfer source and destination data areas need to be defined by creating a set of Linked Lists first.

Each setting is called LLI (LinkedList).

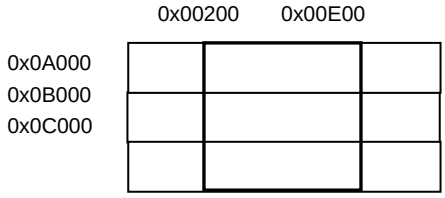
Each LLI controls the transfer of one block of data. Each LLI indicates normal DMA setting and controls transfer of successive data. Each time each DMA transfer is complete, the next LLI setting will be loaded to continue the DMA operation (Daisy Chain).

An example of the setting is shown below.

1. The first DMA transfer setting should be made directly in the DMA register.
2. The second and subsequent DMA transfer settings should be written in the addresses of the memory set in "next LLI AddressX."
3. To stop up to N'th DMA transfer, set "next LLI AddressX" to 0x00000000.

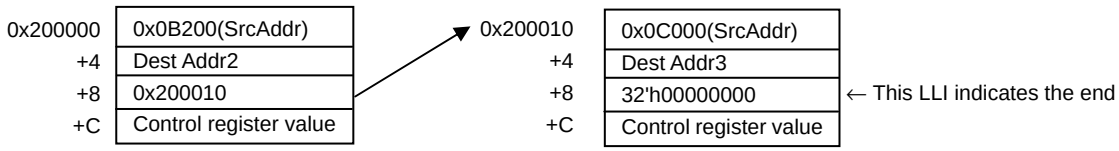


Example: Setting example to transfer the area enclosed by the square in the left figure.



DMACCxSrcAddr: 0x0A200
DMACCxDestAddr: Destination address 1
DMACCxLLI: 0x200000
DMACCxControl: Set the number of burst transfers and the number of transfers, etc.

Linked List



23 Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

23.1 Flash Memory

23.1.1 Features

1) Memory capacity
The TMPM380/382 contains flash memory. The memory sizes and configurations are shown in the table below. Independent write access to each block is available. When the CPU is to access the internal flash memory, 32-bit data bus width is used.

2) Write/erase time
Writing is executed per page. The TMPM380/382 contains 64 words in a page.
Page writing requires 1.25ms (typical) regardless of number of words.
A block erase requires 0.1 s. (typical).
The following table shows write and erase time per chip.

Product Name	Memory Size	Block Configuration				# of Words	Write Time	Erase Time
		128KB	64KB	32KB	16KB			
TMPM380FY	256KB	0	3	1	2	64	1.28s	0.4s
TMPM380/382FW	128KB	0	1	1	2	64	1.28s	0.4s
TMPM382FS	64KB	0	0	1	2	64	1.28s	0.4s

(Note) The above values are theoretical values not including data transfer time.
The write time per chip depends on the write method to be used by the user.

3) Programming method
The onboard programming mode is available for the user to program (rewrite) the device while it is mounted on the user's board.

・The onboard programming mode

3-1) User boot mode
The user's original rewriting method can be supported.

3-2) Single boot mode
The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

Rewriting method
The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if the user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.
This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. See chapter 24 for details of ROM protection and security function.

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function

23.1.2 Block Diagram of the Flash Memory Section

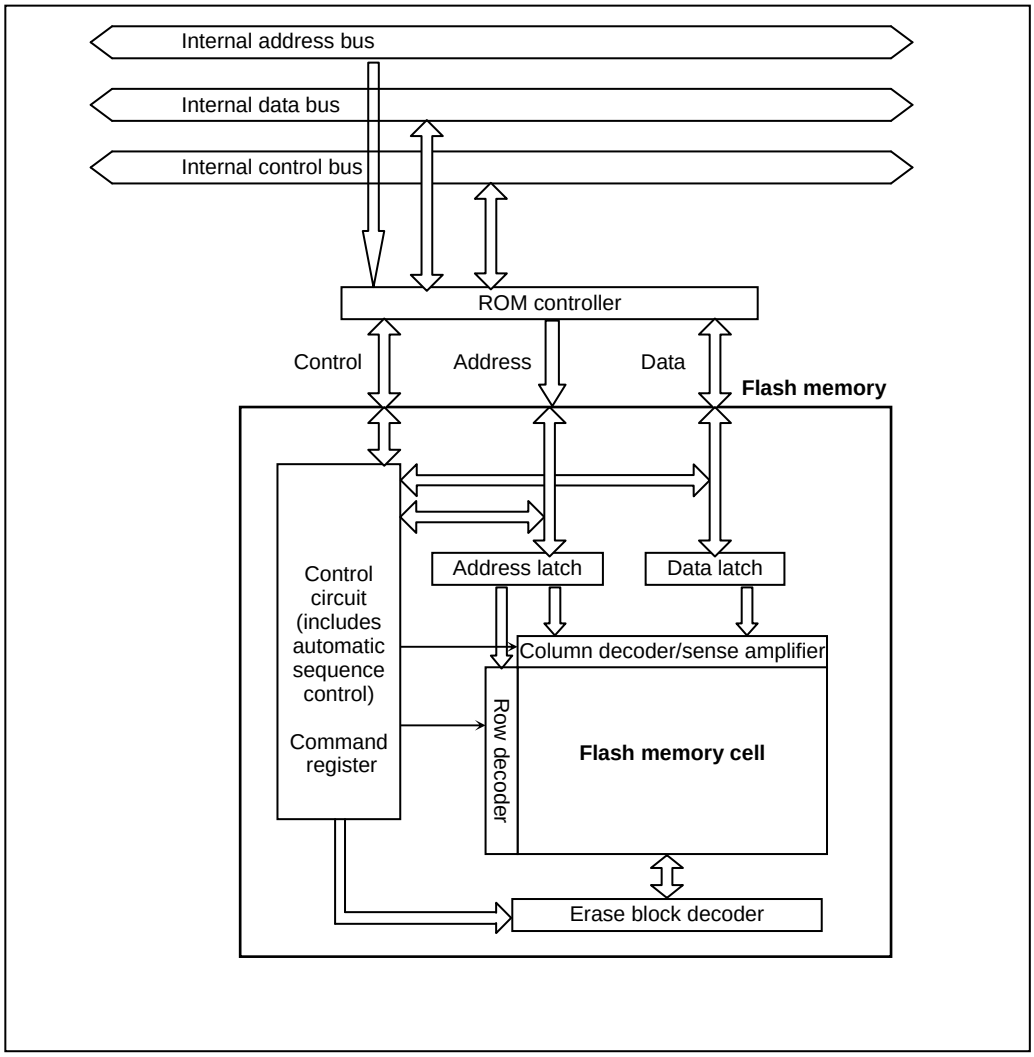


Fig 23-1 Block Diagram of the Flash Memory Section

23.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 23-1 Operation Modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode.
	User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\mathrm{BOOT}}$ (PL0) pin while the device is in reset status.

After the level is set, the CPU starts operation in the selected operation mode when the reset condition is removed. Regarding the $\overline{\mathrm{BOOT}}$ (PL0) pin, be sure not to change the levels during operation once the mode is selected.

The mode setting method and the mode transition diagram are shown below:

Table 23-2 Operation Mode Setting

Operation mode	Pin	
	$\overline{\mathrm{RESET}}$	$\overline{\mathrm{BOOT}}$ (PL0)
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

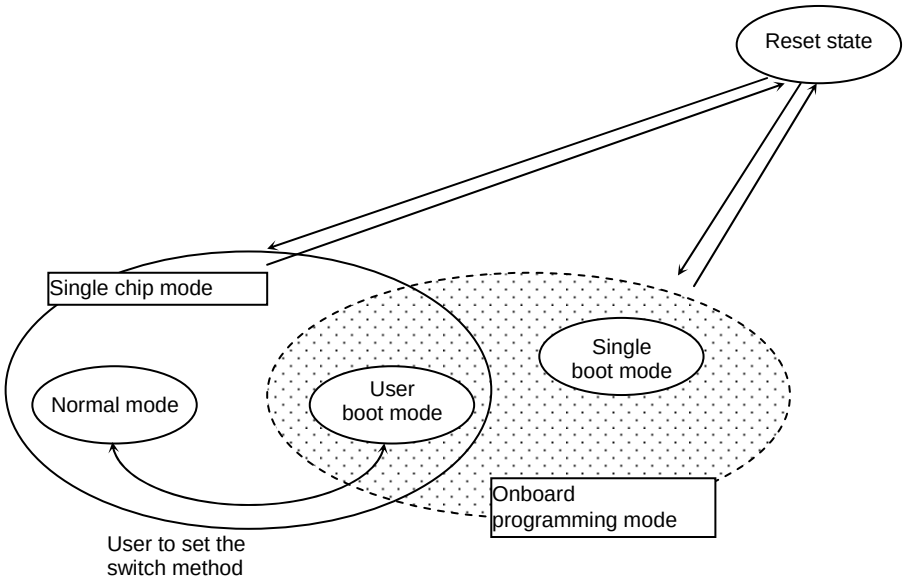


Fig 23-2 Mode Transition Diagram

23.2.1 Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the $\overline{\mathrm{RESET}}$ input is held at "0" for a minimum duration of 12 system clocks (0.3μs with 40MHz operation; the "1/1" clock gear mode is applied after reset).

(Note 1) Regarding power-on reset of devices with internal flash memory;
for devices with internal flash memory, it is necessary to apply "0" to the $\overline{\mathrm{RESET}}$ inputs upon power on for a minimum duration of 300 microseconds regardless of the operating frequency.

(Note 2) While flash auto programming or deletion is in progress, at least 0.5 microseconds of reset period is required regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

23.2.2 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of TMPM380/382 in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete/ writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. All the interruption including a non-maskable are inhibited at User Boot Mode.

(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to On-board Programming of Flash Memory (Rewrite/Erase).

User Boot Mode

(1-A) Method 1: Storing a Programming Routine in the Flash Memory

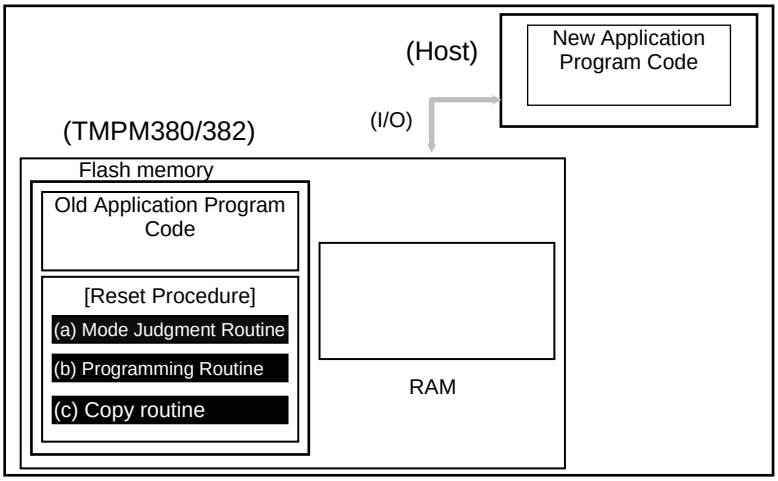
(Step-1)

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM380/382 on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

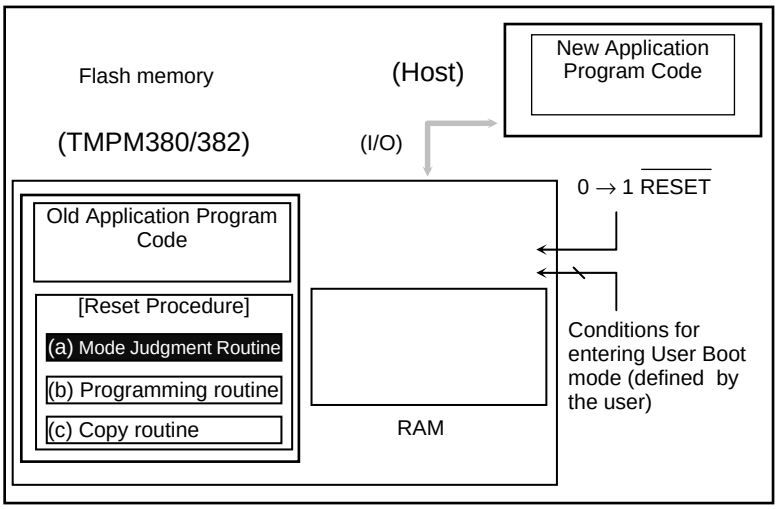
(b) Programming routine: Code to download new program code from a host controller and re-program the flash memory

(c) Copy routine: Code to copy the data described in (b) from the TMPM380/382 flash memory to either the TMPM380/382 on-chip RAM or external memory device.



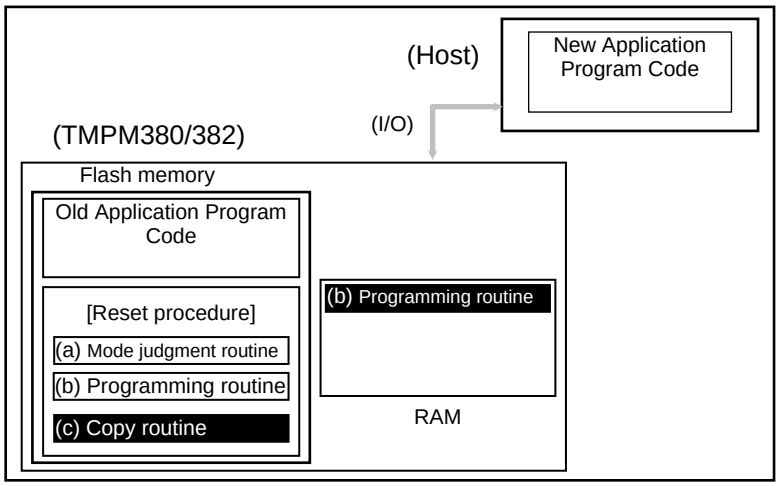
(Step-2)

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM380/382 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode.)



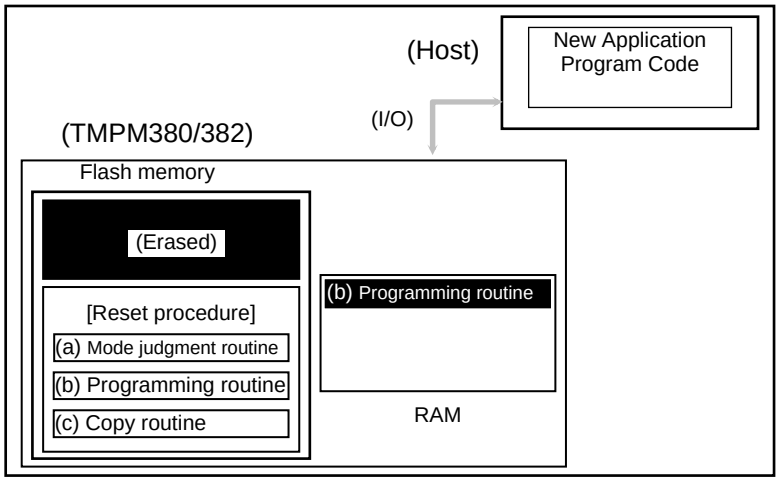
(Step-3)

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM380/382 on-chip RAM.



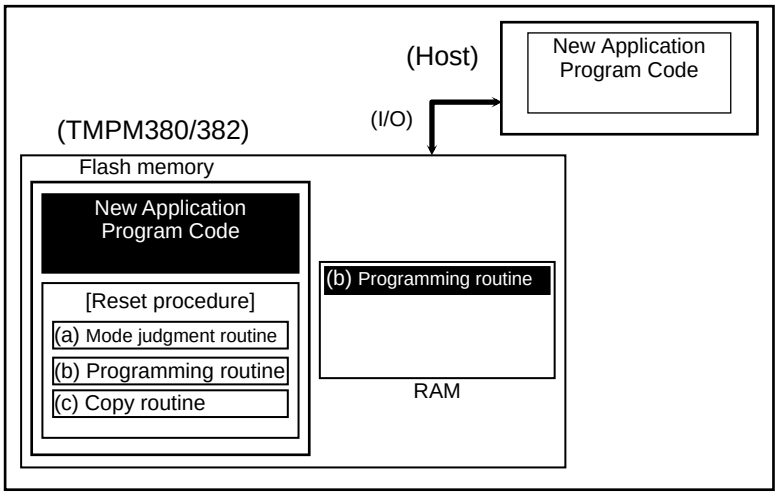
(Step-4)

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



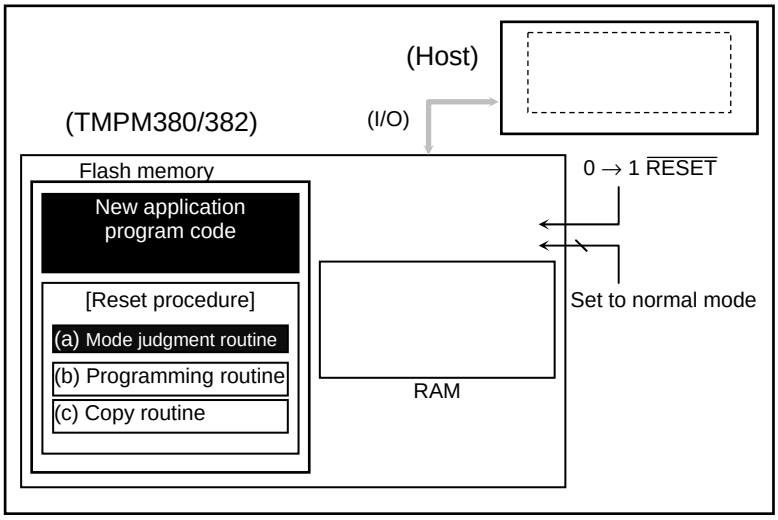
(Step-5)

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(Step-6)

Set $\overline{\text{RESET}}$ to "0" to reset the TMPM380/382. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



(1-B) Method 2: Transferring a Programming Routine from an External Host

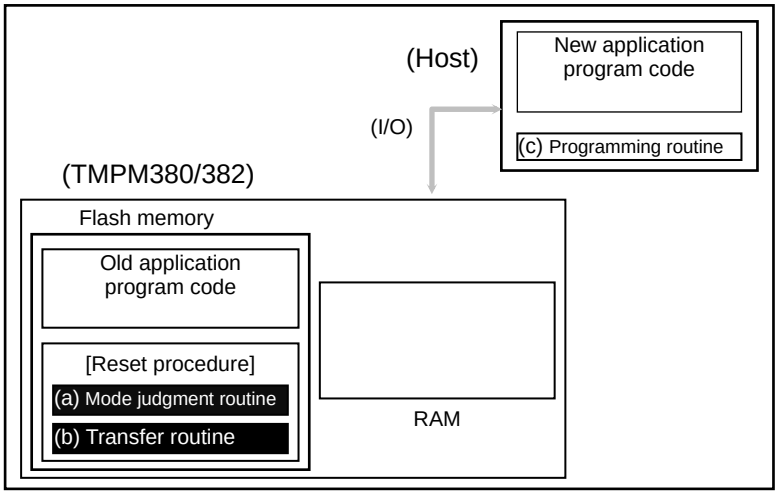
(Step-1)

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM380/382 on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode
(b) Transfer routine: Code to download new program code from a host controller

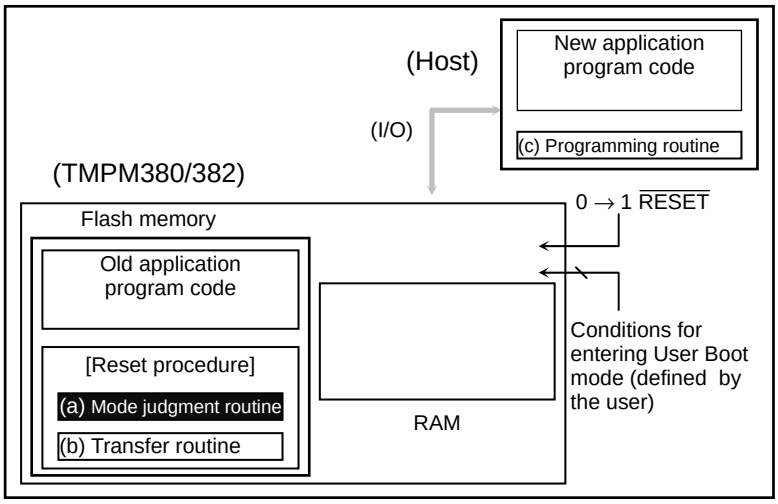
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory



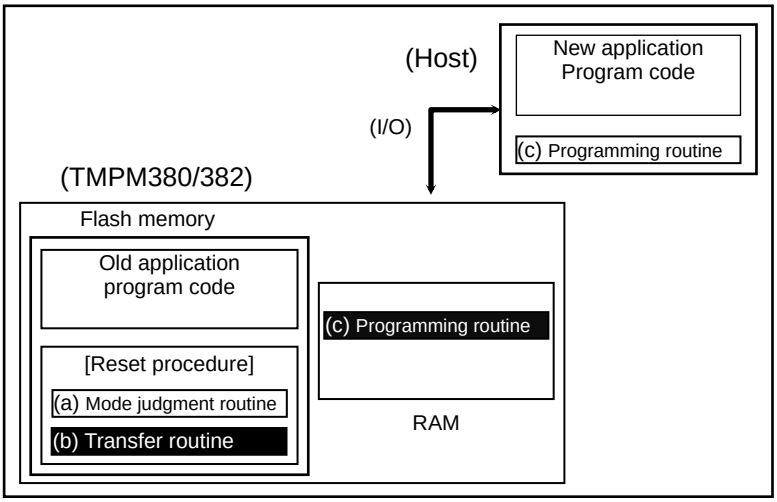
(Step-2)

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM380/382 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode).



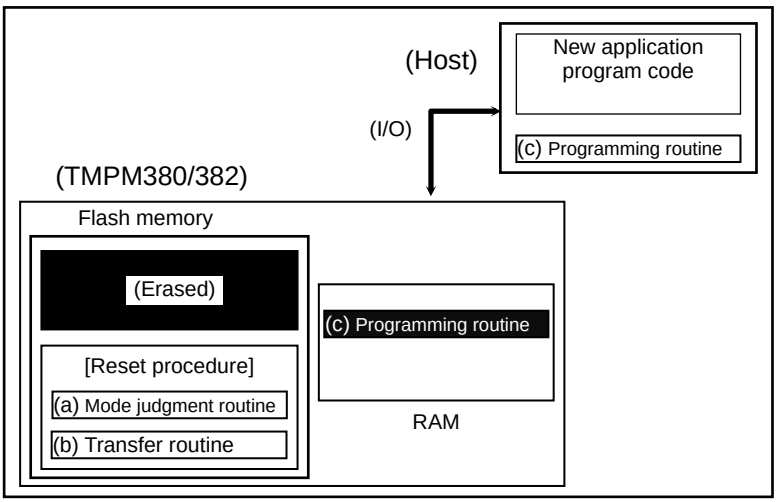
(Step-3)

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to the TMPM380/382 on-chip RAM.



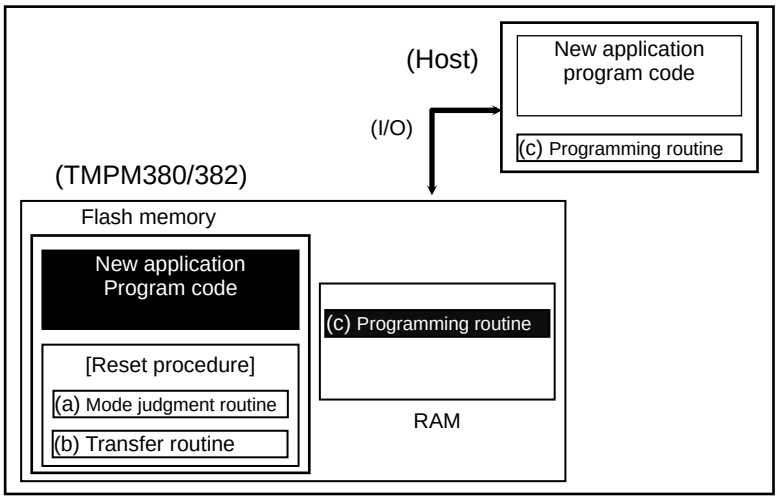
(Step-4)

Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block containing the old application program code.



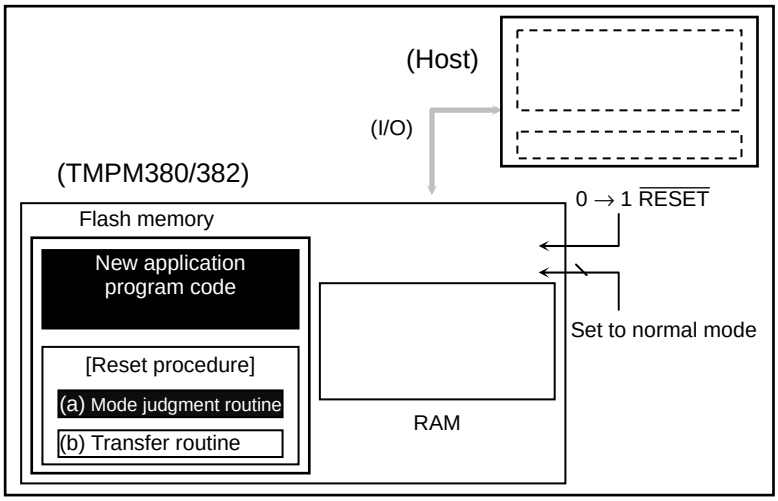
(Step-5)

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(Step-6)

Set $\overline{\text{RESET}}$ to "0" low to reset the TMPM380/382. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



23.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMPM380/382 on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it.

Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMPM380/382 is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMPM380/382 on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory.

Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is checked before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted.

As in the case of User Boot mode, all interrupts including the non-maskable interrupt (NMI) must be disabled in Single Boot mode while the flash memory is being erased or programmed. In Single Boot mode, the boot-ROM programs are executed in Normal mode.

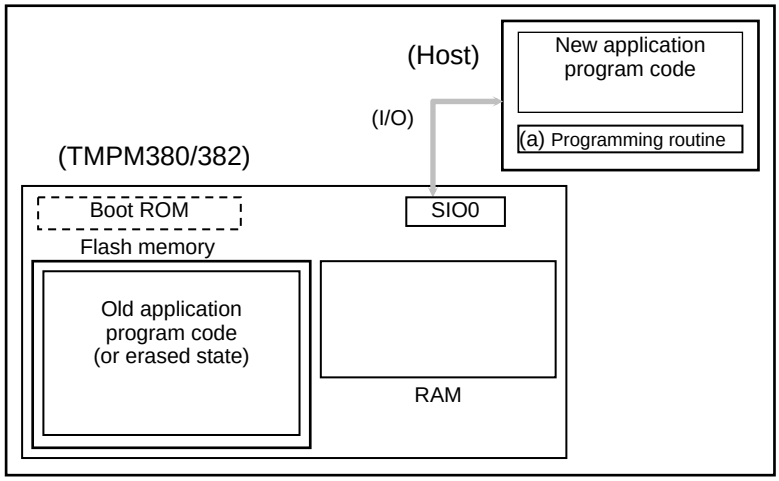
Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations.

Single Boot Mode

(2-A) Using the Program in the On-Chip Boot ROM

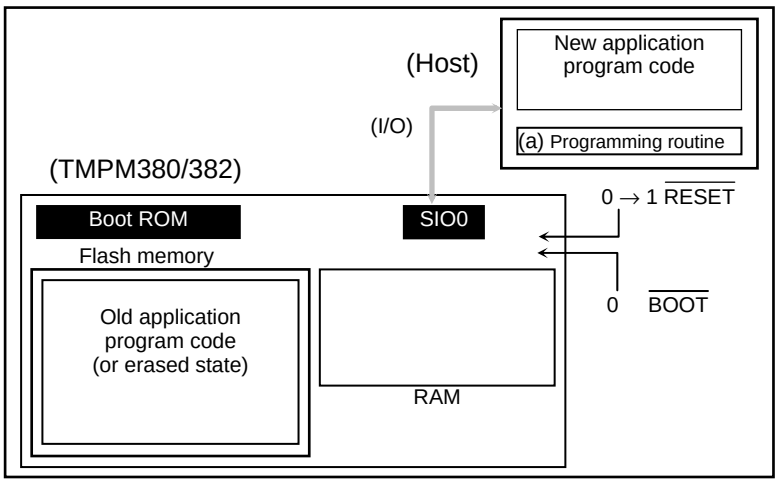
(Step-1)

The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO (SIO0), the SIO0 must be connected to a host controller. Prepare a programming routine (a) on the host controller.



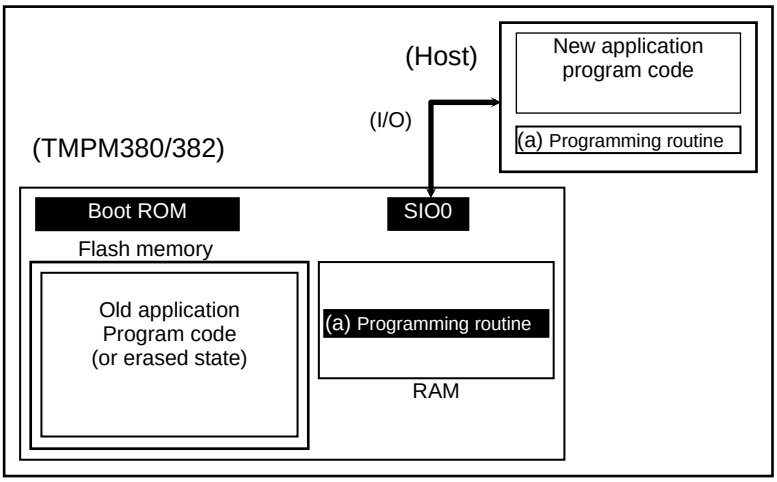
(Step-2)

Cancel the reset of the TMPM380/382 by setting the Single Boot mode pin to “0”, so that the CPU re-boots from the on-chip boot ROM. The 12-byte password transferred from the host controller via SIO0 is first compared to the contents of the special flash memory locations. (If the flash block has already been erased, the password is 0xFF).



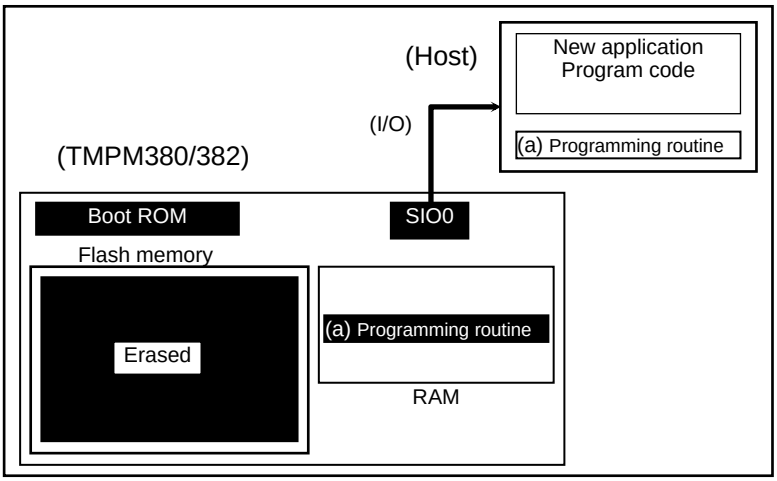
(Step-3)

If the password was correct, the boot program downloads, via the SIO0, the programming routine (a) from the host controller into the on-chip RAM of the TMPM380/382. The programming routine must be stored in the address range 0x2000_0400 to the end address of RAM.



(Step-4)

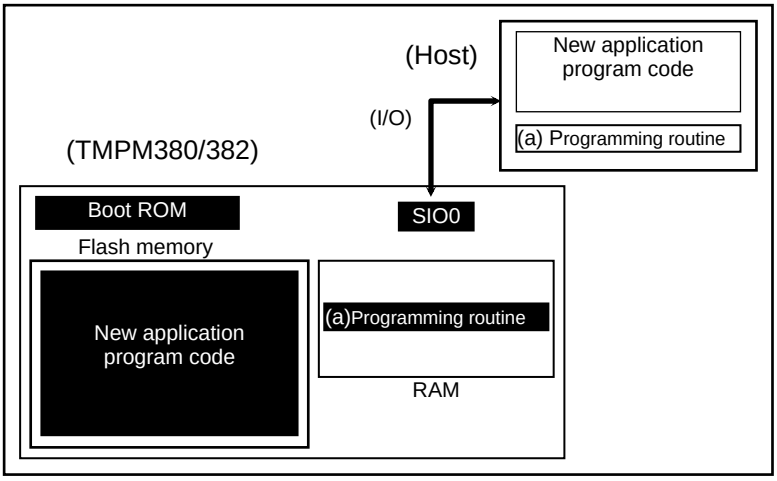
The CPU jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



(Step-5)

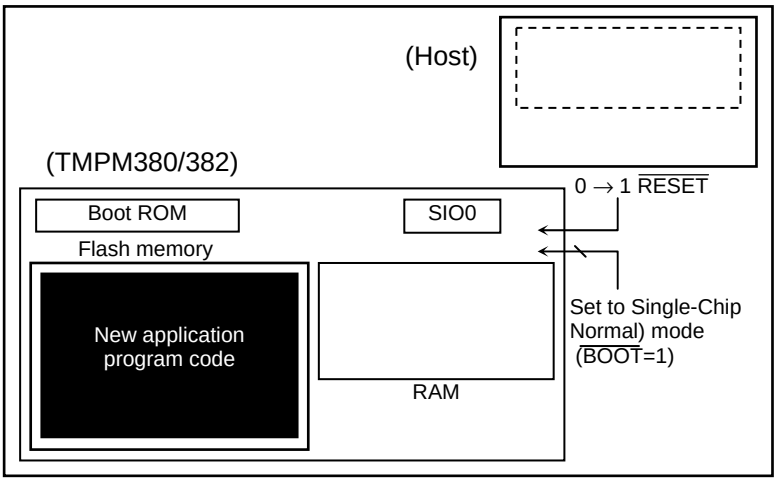
Next, the programming routine (a) downloads new application program code from the host controller and programs it into the erased flash block. Once programming is complete, protection of that flash block is turned on. It is not allowed to move program control from the programming routine (a) back to the boot ROM.

In the example below, new program code comes from the same host controller via the same SIO0 channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(Step-6)

When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMPM380/382 re-boots in Single-Chip (Normal) mode to execute the new program.



(1) Configuration for Single Boot Mode

To execute the on-board programming, boot the TMPM380/382 with Single Boot mode following the configuration shown below.

$\overline{\text{BOOT}}$ (PL0) = 0

$\overline{\text{RESET}}$ = 0 → 1

Set the $\overline{\text{RESET}}$ input to 0, and set the each $\overline{\text{BOOT}}$ (PL0) pins to values shown above, and then release RESET (high).

(2) Memory Map

Fig 23-3 shows a comparison of the memory maps in Normal and Single Boot modes. In Single Boot mode, the internal flash memory is mapped from 0x3F80_0000, and the internal boot ROM (Mask ROM) is mapped to 0x0000_0000 through 0x0000_0FFF.

Product Name	Flash Size	RAM Size	Flash Address (Single Chip/ Single Boot Mode)	RAM Address
TMPM380FY	256KB	16KB	0x0000_0000 - 0x0003_FFFF 0x3F80_0000 - 0x3F83_FFFF	0x2000_0000 - 0x2000_3FFF
TMPM380/382FW	128KB	12KB	0x0000_0000 - 0x0001_FFFF 0x3F80_0000 - 0x3F81_FFFF	0x2000_0000 - 0x2000_2FFF
TMPM382FS	64KB	8KB	0x0000_0000 - 0x0000_FFFF 0x3F80_0000 - 0x3F80_FFFF	0x2000_0000 - 0x2000_1FFF

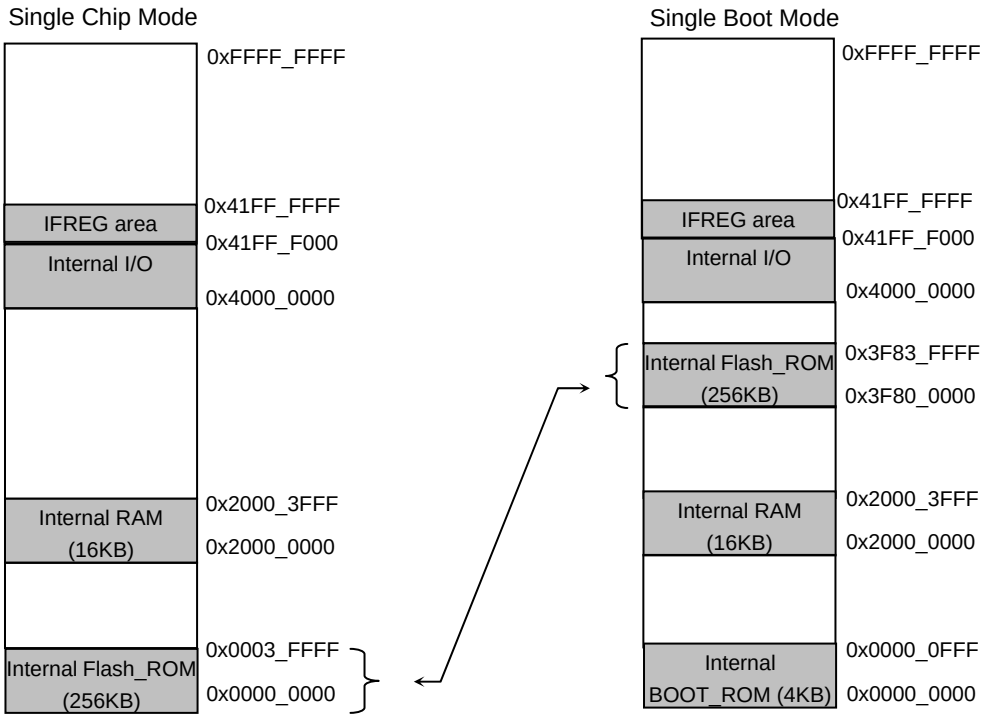


Fig 23-3 Memory Maps for TMPM380FY

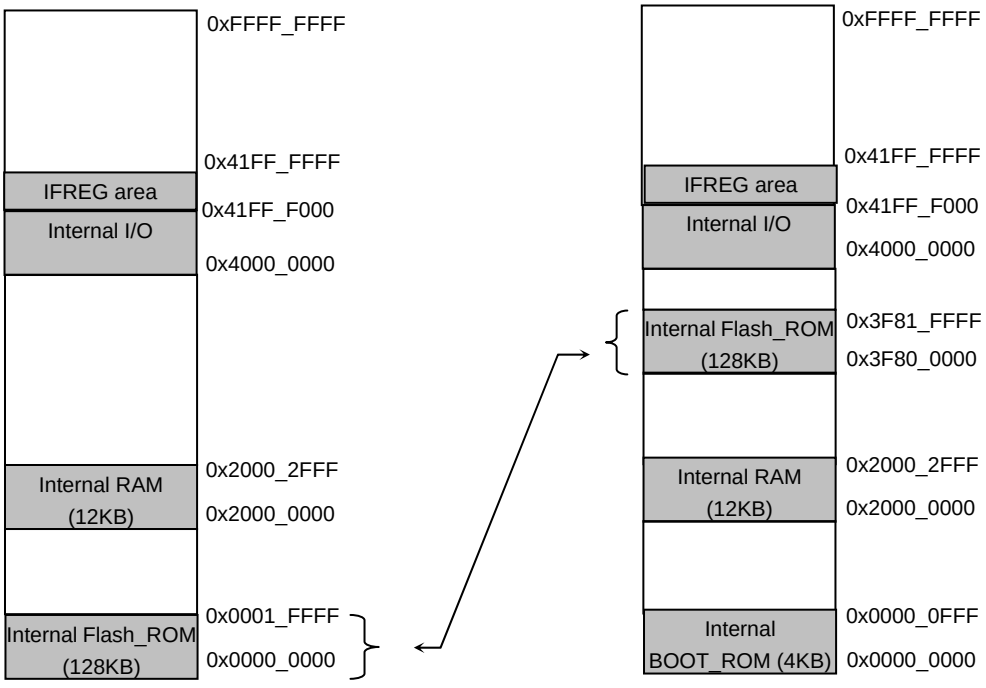


Fig 23-4 Memory Maps for TMPM380/382FW

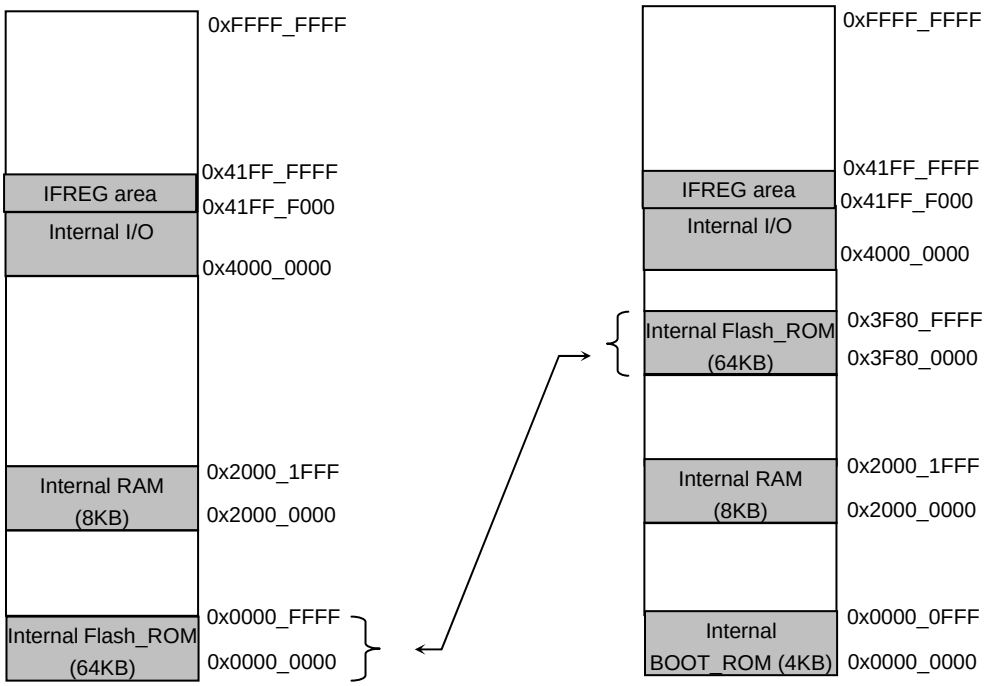


Fig 23-5 Memory Maps for TMPM382FS

(3) Interface specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. The same configuration is applied to a communication format on a programming controller to execute the on-board programming. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below.

- UART communication

  Communication channel : SIO channel 0
  Serial transfer mode : UART (asynchronous), half -duplex, LSB first
  Data length : 8 bit
  Parity bits : None
  STOP bits : 1 bit
  Baud rate : Arbitrary baud rate

- I/O interface mode

  Communication channel : SIO channel 0
  Serial transfer mode : I/O interface mode, full -duplex, LSB first
  Synchronization clock (SCLK0) : Input mode
  Handshaking signal : PE4 configured as an output mode
  Baud rate : Arbitrary baud rate

Table 23-3 Required Pin Connections

Pins		Interface	
		UART	I/O Interface Mode
Power supply pins	DVDD5	○	○
	REGVDD5	○	○
	AVDD	○	○
	VOUT3	○	○
	DVSS	○	○
	AVSS	○	○
	CVSS	○	○
Mode-setting pin	$\overline{\text{BOOT}}$ (PL0)	○	○
Reset pin	$\overline{\text{RESET}}$	○	○
Communication pins	TXD0(PE0)	○	○
	RXD0(PE1)	○	○
	SCLK0(PE2)	x	○ (Input mode)
	PE4	x	○ (Output mode)

(4) Data Transfer Format

Table 23-4, Table 23-6 and Table 23-7 illustrate the operation commands and data transfer formats at each operation mode. In conjunction with this section, refer to (6) Operation of Boot Program.

Table 23-4 Single Boot Mode Commands

Code	Command
0x10	RAM transfer
0x20	Don't care
0x30	Don't care
0x40	Chip and protection bit erase

Note : Code 0x20 and 0x30 are use for internal test

(5) Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 23-5.

Table 23-5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	BOOT ROM is mapped from 0x2000_0000 to 0x2000_03FF. Store the RAM transfer program from 0x2000_0400 through the end address of RAM.
Internal ROM	The following addresses are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable. TMPM380FY : 0x3F83_FFF0 to 0x3F83_FFFF TMPM380/382FW : 0x3F83_FFF0 to 0x3F83_FFFF TMPM382FS : 0x3F83_FFF0 to 0x3F83_FFFF

Table 23-6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMPM380/382	Baud rate	Data Transferred from the TMPM380/382 to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode 0x86 For I/O Interface mode 0x30	Desired baud rate (Note 1)	-
	2 byte	-		ACK for the serial operation mode byte For UART mode -Normal acknowledge 0x86 (The boot program aborts if the baud rate can not be set correctly.) For I/O Interface mode -Normal acknowledge 0x30
	3 byte	Command code (0x10)		-
	4 byte	-		ACK for the command code byte (Note 2) -Normal acknowledge 0x10 -Negative acknowledge 0xN1 -Communication error 0xN8
	5 byte - 16 byte	Password sequence (12 bytes) 0x3F83_FFF4 to 0x3F83_FFFF		-
	17 byte	Check SUM value for bytes 5 - 16		-
	18 byte	-		ACK for the checksum byte (Note 2) -Normal acknowledge 0xN0 -Negative acknowledge 0xN1 -Communication error 0xN8
	19 byte	RAM storage start address 31 - 24		-
	20 byte	RAM storage start address 23 - 16		-
	21 byte	RAM storage start address 15 - 8		-
	22 byte	RAM storage start address 7 - 0		-
	23 byte	RAM storage byte count 15 - 8		-
	24 byte	RAM storage byte count 7 - 0		-
	25 byte	Check SUM value for bytes 19 - 24		-
	26 byte	-		ACK for the checksum byte (Note 2) -Normal acknowledge 0xN0 -Negative acknowledge 0xN1 -Communication error 0xN8
	27 byte ~ m byte	RAM storage data		-
	m + 1 byte	Checksum value for bytes 27 - m		-
	m + 2 byte	-		ACK for the checksum byte (Note 2) -Normal acknowledge 0xN0 -Negative acknowledge 0xN1 -Communication error 0xN8
RAM	m + 3 byte	-		Jump to RAM storage start address

(Note 1) In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

(Note 2) In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur. "N" in the error code shows an upper 4 bits [7:4] of command code in the 3rd byte. For example, when a password error occurs in RAM Transfer Command, the ACK is "0x11".

(Note 3) The 19th to 25th bytes must be within the RAM address range from 0x2000_0400 through the end address of RAM.

Table 23-7 Transfer Format for the Chip and Protection Bit Erase Command

	Byte	Data Transferred from the Controller to the TMPM380/382	Baud rate	Data Transferred from the TMPM380/382 to the Controller
Boot ROM	1 byte	Serial operation mode and baud rate For UART mode 0x86 For I/O Interface mode 0x30	Desired baud rate (Note 1)	-
	2 byte	-		ACK for the serial operation mode byte For UART mode -Normal acknowledge 0x86 For I/O Interface mode -Normal acknowledge 0x30 (The boot program aborts if the baud rate can not be set correctly.)
	3 byte	Command code (0x40)		-
	4 byte	-		ACK for the command code byte (Note 2) -Normal acknowledge 0x40 -Negative acknowledge 0xN1 -Communication error 0xN8
	5 byte	Chip erase command code (0x54)		-
	6 byte	-		ACK for the command code byte (Note 2) -Normal acknowledge 0x54 -Negative acknowledge 0xN1 -Communication error 0xN8
	7 byte	-		ACK for the chip erase command code byte -Normal acknowledge 0x4F -Negative acknowledge 0x4C
	8 byte	(Wait for the next command code.)		-

(Note 1) In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

(Note 2) In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur. "N" in the error code shows an upper 4 bits [7:4] of command code in the 3rd byte. For example, when a password error occurs in Chip and Protection Bit Erase Command, the ACK is "0x41".

(6) Operation of Boot Program

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these two commands, of which the details are provided on the following subsections. The addresses described in this section are the virtual unless otherwise noted.

1. RAM Transfer command

The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The user program RAM space can be assigned to the range from 0x2000_0400 to the end address of RAM, whereas the boot program area (0x2000_0000 to 0x2000_03FF) is unavailable. The user program starts at the assigned RAM address.

The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section20.3.

Before initiating a transfer, the RAM Transfer command verifies a password sequence coming from the controller against that stored in the flash memory.

2. Chip and Protection Bit Erase command

This command erases the entire area of the flash memory automatically without verifying a password. All the blocks in the memory cell and their protection conditions are erased even when any of the blocks are prohibited from writing and erasing. When the command is completed, the SECBIT <SECBIT> bit is set to “1”.

This command serves to recover boot programming operation when a user forgets the password. Therefore password verification is not executed.

1) RAM Transfer Command (See Table 23-6)

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see Determination of a Serial Operation Mode described later. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the SC0MOD0 register is cleared.

   - To communicate in UART mode
     Send, from the controller to the target board, 86H in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the first byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.

   - To communicate in I/O Interface mode
     Send, from the controller to the target board, 0x30 in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

     In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip's operating frequency is high, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate. When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the third byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (bit 3, 0xN8).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the first byte: 0x86 for UART mode and 0x30 for I/O Interface mode.

   UART mode
   If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the SC0BRCR and sends back 0x86 to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication. Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 0x86 within the allowed time-out period, the controller should give up the communication. The boot program sets the RXE bit in the SC0MOD0 register to enable reception (1) before loading the SIO transmit buffer with 0x86.

   - I/O Interface mode
     The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 0x30 to

the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 0x30, then the controller should take it as a go-ahead. The controller must then deliver the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD register to enable reception before loading the SIO transmit buffer with 0x30.

3. The 3rd byte transmitted from the controller to the target board is a command. The code for the RAM Transfer command is 0x10.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (bit 3) and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 23-4, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 0x10 and then branches to the RAM Transfer routine. Once this branch is taken, password verification is done. Password verification is detailed in a later section “Password”. If the 3rd byte is not a valid command, the boot program sends back 0xN1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes transmitted from the controller to the target board, are a 12-byte password. Each byte is compared to the contents of following addresses in the flash memory. The verification is started with the 5^{th} byte and the smallest address in the designated area. If the password verification fails, the RAM Transfer routine sets the password error flag.

Product name	Area
TMPM380FY	0x3F83_FFF4 – 0x3F83_FFFF
TMPM380/382FW	
TMPM382FS	

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two’s complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section “Checksum Calculation”.

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge

response to the 5th to 17th bytes. First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 18H (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

Finally, the RAM Transfer routine examines the result of the password verification. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- Irrespective of the result of the password comparison, all the 12 bytes of a password in the flash memory are the same value other than 0xFF.
Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above verification has been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

8. The 19th to 22nd bytes, transmitted from the controller the target board, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31–24 of the address and the 22nd byte corresponds to bits 7–0 of the address.

9. The 23rd and 24th bytes, transmitted from the controller to the target board, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15–8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7–0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two’s complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section “Checksum Calculation”.

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 0x18 and returns to the command wait state (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 25th bytes must result in 00H (with the carry dropped). If it is not 00H, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- The RAM storage start address must be within the range of 0x2000_0400 to the end address of RAM.

When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMPM380/382. Storage begins at the address specified by the 19th–22nd bytes and continues for the number of bytes specified by the 23rd–24th bytes.

13. The (m+1) th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in a later section "Checksum Calculation".

14. The (m+2) th byte is a acknowledge response to the 27th to (m+1) th bytes.

    First, the RAM Transfer routine checks for a receive error in the 27th to (m+1) th bytes. If there was a receive error, the RAM Transfer routine sends back 18H (bit 3) and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

    Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m+1) th bytes must result in 0x00 (with the carry dropped). If it is not 0x00, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 0x11 (bit 0) to the controller and returns to the command wait state (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (0x10) to the controller.

15. If the (m+2) th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes.

2) Chip and Protection Bit Erase command (See Table 23-7)

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x40.

3. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

   If the 3rd byte is equal to any of the command codes listed in Table 23-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x40. If the 3rd byte is not a valid command, the boot program sends back 0xN1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The 5th byte, transmitted from the target board to the controller, is the Chip Erase Enable command code (0x54).

5. The 6th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th byte.
   Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xN8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.
   If the 5th byte is equal to any of the command codes to enable erasing, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x54 and then branches to the Chip Erase routine. If the 5th byte is not a valid command, the boot program sends back 0xN1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

6. The 7th byte indicates whether the Chip Erase command is normally completed or not.
   At normal completion, completion code (0x4F) is sent.
   When an error was detected, error code (0x4C) is sent.

7. The 9th byte is the next command code.

5) Acknowledge Responses

The boot program represents processing states with specific codes. Table 23-8 to Table 23-11 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always 0. Receive error checking is not done in I/O Interface mode.

Table 23-8 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

(Note) If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.

Table 23-9 ACK Response to the Command Byte

Return Value	Meaning
0xN8 (See Note)	A receive error occurred while getting a command code.
0xN1 (See Note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.
0x20	Command was received.
0x30	Command was received.
0x40	The Chip Erase command was received.

(Note1) The upper four bits of the ACK response are the same as those of the previous command code.

(Note2) Command 0x20 and 0x30 are use for internal test only.

Table 23-10 ACK Response to the Checksum Byte

Return Value	Meaning
0xN8 (See Note)	A receive error occurred.
0xN1 (See Note)	A checksum or password error occurred.
0xN0 (See Note)	The checksum was correct.

(Note) The upper four bits of the ACK response are the same as those of the operation command code. It is 1 (N＝RAM transfer command data [7:4]) when password error occurs.

Table 23-11 ACK Response to Chip and Protection Bit Erase Byte

Return Value	Meaning
0x54	The Chip Erase enabling command was received.
0x4F	The Chip Erase command was completed.
0x4C	The Chip Erase command was abnormally completed.

6) Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 0x86 at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 0x30 at 1/16 the desired baud rate. Fig 23-6 shows the waveforms for the first byte.

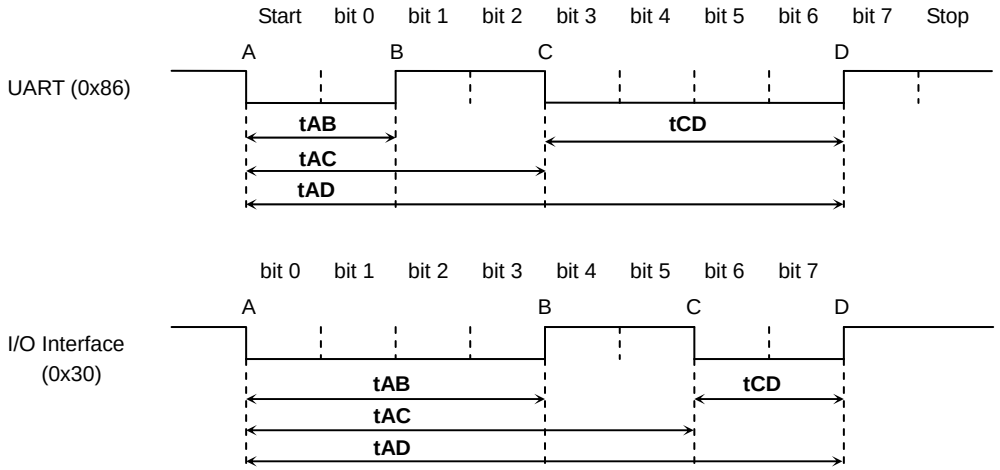


Fig 23-6 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Table 23-5 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Table 23-5 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (0x86) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 0x30, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

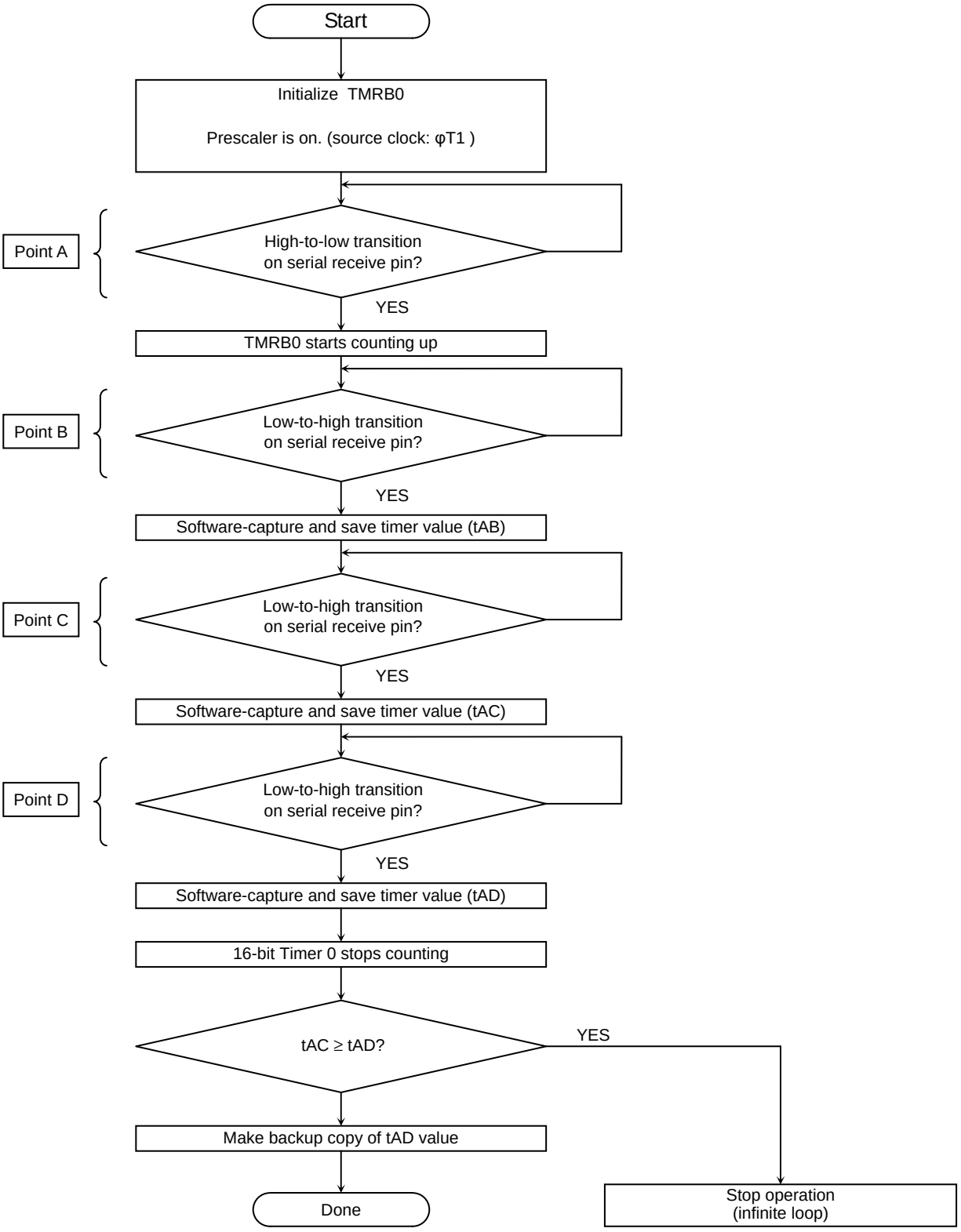


Fig 23-7 Serial Operation Mode Byte Reception Flow

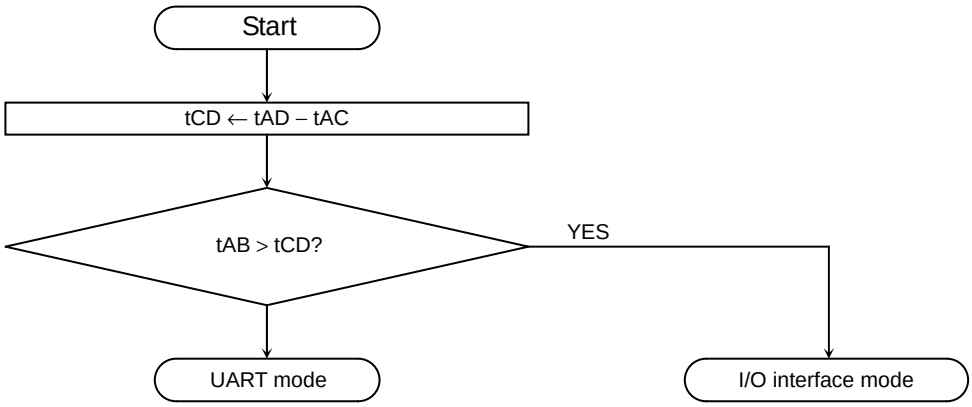


Fig 23-8 Serial Operation Mode Determination Flow

7) Password

The RAM Transfer command (0x10) causes the boot program to perform password verification. Following an echo-back of the command code, the boot program verifies the contents of the 12-byte password area within the flash memory. The following table shows the password area.

Product name	Area
TMPM380FY	0x3F83_FFF4 – 0x3F83_FFFF
TMPM380/382FW	
TMPM382FS	

If all these address locations contain the same bytes of data other than 0xFF, a password area error occurs as shown in Fig 23-9. In this case, the boot program returns an error acknowledge (0x11) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all 0xFFs.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password verification. Otherwise, a password error occurs, which causes the boot program to reply an error acknowledge in response to the checksum byte (the 17th byte).

The password verification is performed even if the security function is enabled.

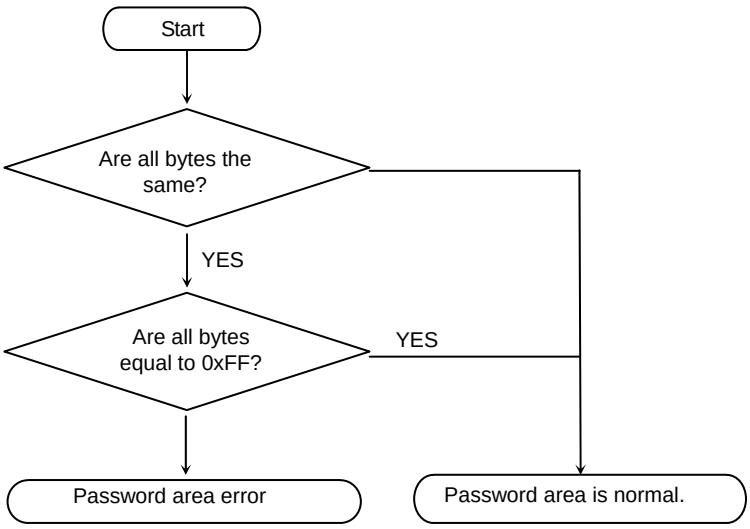


Fig 23-9 Password Area Verification Flow

8) Checksum Calculation

The checksum byte for a series of bytes of data is calculated by adding the bytes together, dropping the carries, and taking the two's complement of the total sum. The controller must perform the same checksum operation in transmitting checksum bytes.

Example) Assume the upper and lower bytes of the sum as 0xE5 and 0xF6. To calculate the checksum for a series of 0xE5 and 0xF6:

Add the bytes together

0xE5 + 0xF6 = 0x1DB

Take the two's complement of the sum, and that is the checksum byte.

0 – 0xDB = 0x25

(7) General Boot Program Flowchart

Fig 23-10 shows an overall flowchart of the boot program.

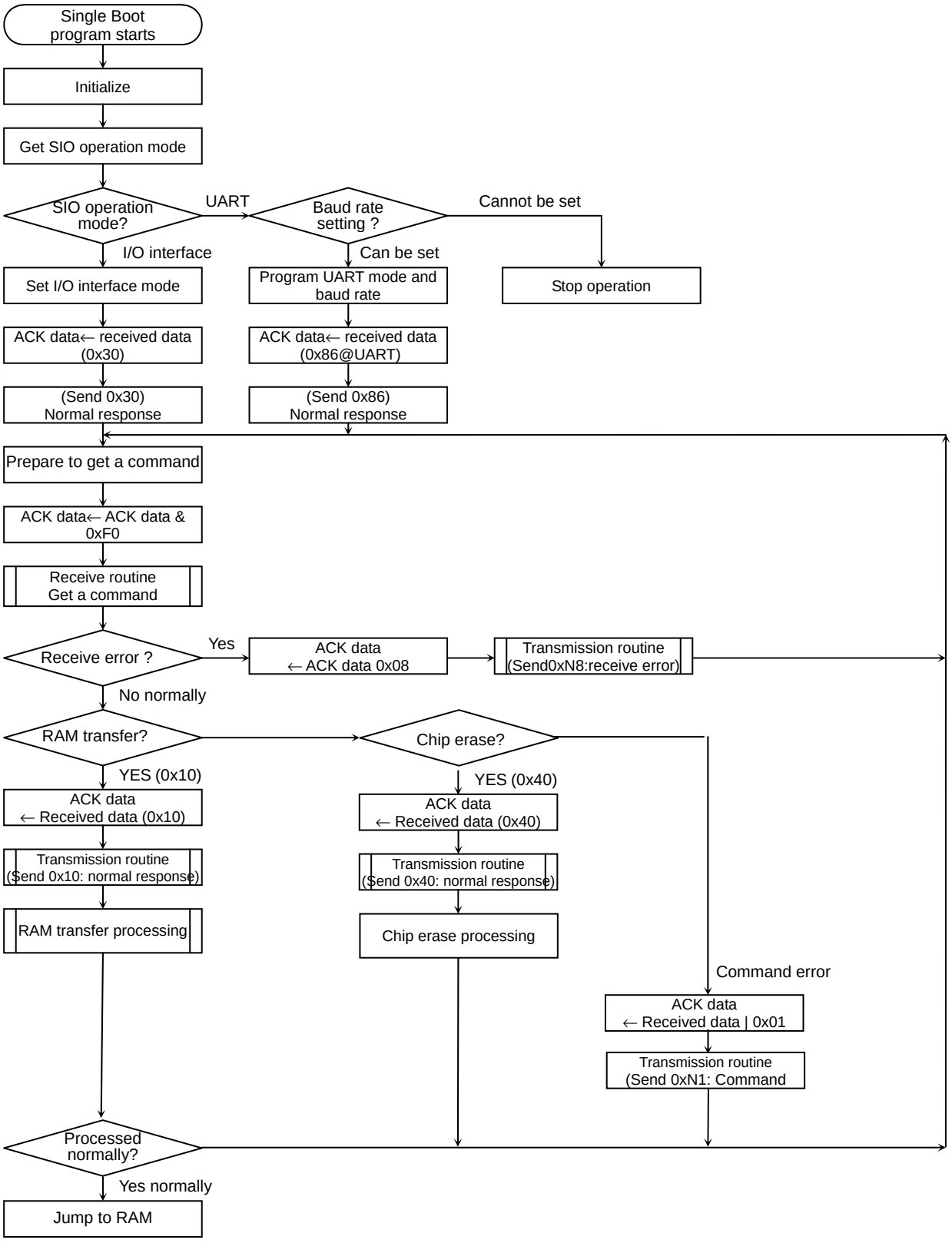


Fig 23-10 Overall Boot Program Flow

23.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM or from an external memory device after shifting to the user boot mode.

23.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands. In writing or erasing, use 32-bit data transfer command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 23-12 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically per page.
Automatic chip erase	Erases the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Protect function	By writing a 4-bit protection code, the write or erase function can be individually inhibited for each block.

Note that addressing of operation commands is different from the case of standard commands due to the specific interface arrangements with the CPU. Also note that the flash memory is written in 32-bit blocks. So, 32-bit (word) data transfer commands must be used in writing the flash memory.

(1) Block configuration

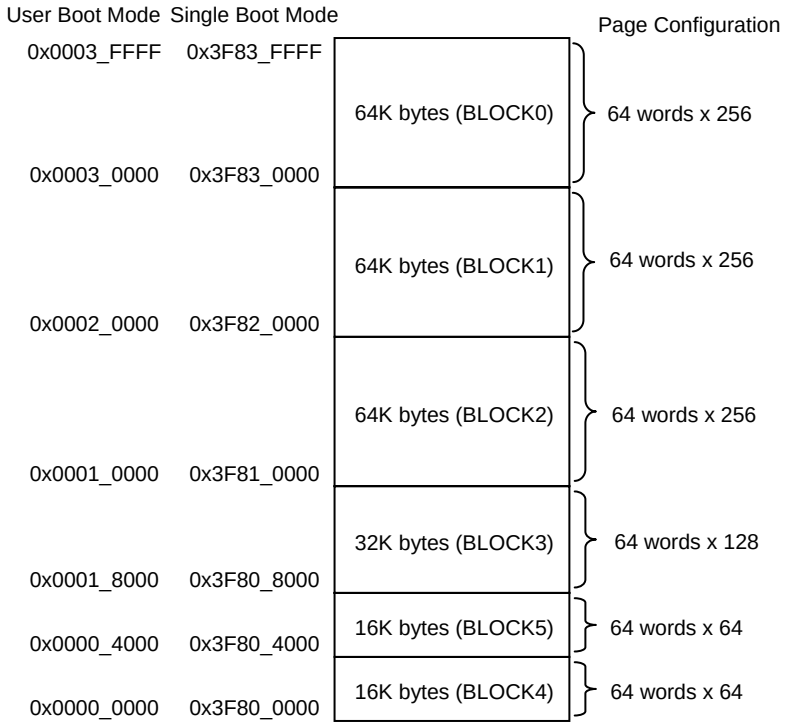


Fig 23-11 Block Configuration of Flash Memory (TMPM380FY)

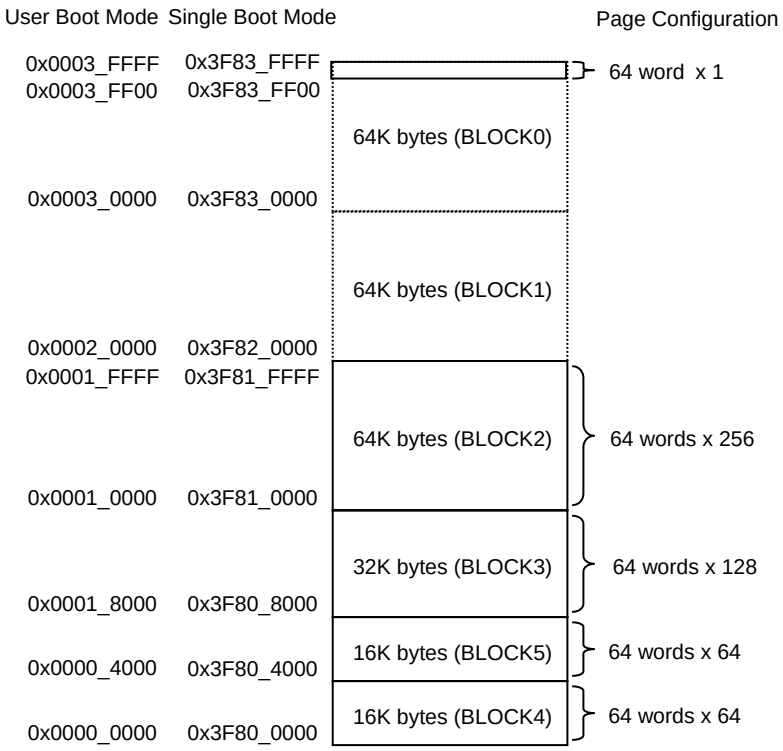


Fig 23-12 Block Configuration of Flash Memory (TMPM380/382FW)

(Note) In addition to 128KB flash area, the TMPM380FW/382FW provides 64-word data/password area (0x3F83_FF00 . 0x3F83_FFFF, 1 page) for Show Product Information command. To erase the content, execute the automatic chip erase command or assign block 0 with the automatic block erase command.Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.

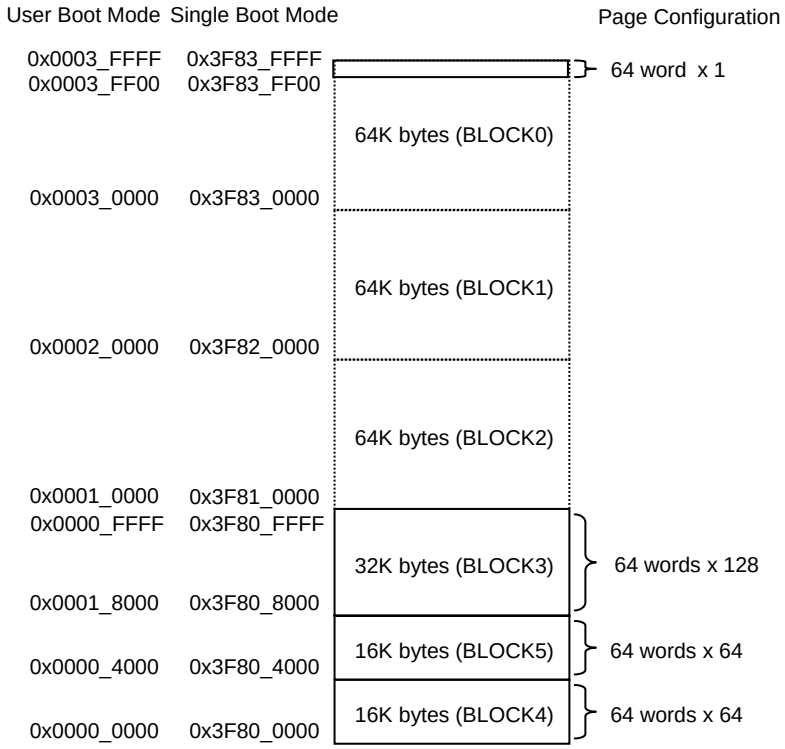


Fig 23-13 Block Configuration of Flash Memory (TMPM382FS)

(Note) In addition to 64KB flash area, the TMPM382FS provides 64-word data/password area (0x3F83_FF00 . 0x3F83_FFFF, 1 page) for Show Product Information command. To erase the content, execute the automatic chip erase command or assign block 0 with the automatic block erase command.Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.

(2) Basic operation

Generally speaking, this flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exceptions other than debug exceptions and reset while a debug port is connected. Any exception generation cannot set the device to the read mode except when a hardware reset is generated.

1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- **Read/reset command and Read command (software reset)**

When ID-Read command is used, the reading operation is terminated instead of automatically returning to the read mode. In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used. The Read command is used to return to the read mode after executing 32-bit data transfer command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.

- **With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.**

2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read

While commands are generally comprised of several bus cycles, the operation to apply 32-bit data transmit command to the flash memory is called "bus write cycle." The bus write cycles are to be in a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of a command write operation is in accordance with a predefined specific sequence. If any bus write cycle does not follow a

predefined command write sequence, the flash memory will terminate the command execution and return to the read mode.

(Note 1) Command sequences are executed from outside the flash memory area.

(Note 2) Each bus write cycle must be sequentially executed by 32-bit data transmit command. While a command sequence is being executed, access to the flash memory is prohibited. Also, don't generate any interrupt (except debug exceptions when a DSU probe is connected).If such an operation is made, it can result in an unexpected read access to the flash memory and the command sequencer may not be able to correctly recognize the command. While it could cause an abnormal termination of the command sequence, it is also possible that the written command is incorrectly recognized.

(Note 3) For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle that the FLCS RDY/BSY bit is set to "1." It is recommended to subsequently execute a Read command.

(Note 4) Upon issuing a command, if any address or data is incorrectly written, be sure to perform a software reset to return to the read mode again.

(3) Reset

Hardware reset

A hardware reset is used to cancel the operational mode set by the command write operation when forcibly termination during auto programming/ erasing or abnormal termination during auto operations occurs. The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the RESET input pin of this device is set to V_{IL} or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section 23.2.1 "Reset Operation" for CPU reset operations. After a given reset input, the CPU will read the reset vector data from the flash memory and starts operation after the reset is removed.

(4) Commands

1) Automatic Page Programming

Writing to a flash memory device is to make "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For making "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data of each page. The TMPM380/382 contains 64 words in a page. A 64 word block is defined by a same [31:8] address and it starts from the address [7:0] = 0 and ends at the address [7:0] = 0xFF. This programming unit is hereafter referred to as a "page."

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by the FLCS [0] <RDY/BSY> register.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more times irrespective of the data cell value whether it is "1" or "0." Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content can cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the third bus write cycle of the command cycle is completed. On and after the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at a time). Be sure to use the 32-bit data transfer command in writing commands on and after the fourth bus cycle. In this, any 32-bit data transfer commands shall not be placed across word boundary. On and after the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0." For example, if the top address of a page is not to be written, set the input data of the fourth bus write cycle to 0xFFFFFFFF to command write the data.

Once the fourth bus cycle is executed, it is in the automatic programming operation. This condition can be checked by monitoring the register bit FLCS [0] <RDY/BSY> (See Table 23-13). Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted.

When a single page has been command written normally terminating the automatic page writing process, the FLCS [0] <RDY/BSY> bit is set to "1" and it returns to the read mode. When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FLCS [0] <RDY/BSY> (Table 23-13). If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to execute hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

(Note) Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.

2) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FLCS [0] <RDY/BSY> (See Table 23-13). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the mode and will not return to the read mode.

For returning to the read mode, it is necessary to execute hardware reset to reset the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

3) Automatic block erase (fro aech block)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FLCS <RDY/BSY> (See Table 23-13). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation

has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, execute hardware reset to reset the device.

4) Automatic programming of protection bits (for each block)
This device is implemented with protection bits. This protection can be set for each block. See Table 23-18 for table of protection bit addresses. This device assigns 1 bit to 1 block as a protection bit. The applicable protection bit is specified by PBA in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by the FLCS <BLPRO> register to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FLCS <RDY/BSY> (See Table 23-13). Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, all the FLCS <BLPRO> bits are set to "1" indicating that it is in the protected state (See Table 23-13). This disables subsequent writing and erasing of all blocks.

(Note) Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. The FLCS <RDY/BSY> bit turns to "0" after entering the seventh bus write cycle.

5) Automatic erasing of protection bits

Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits and the security bits. It depends on the status of FLCS <BLPRO> whether all the <BLPRO> bits are set to "1" or not if SECBIT<SECBIT> is 0x1. Be sure to check the value of FLCS <BLPRO> before executing the automatic protection bit erase command. See chapter 17 for details.

・When all the FLCS <BLPRO> bits are set to "1" (all the protection bits are programmed):

When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FLCS will be set to "0x00000001." While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FLCS <BLPRO> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

・When the FLCS <BLPRO> bits include "0" (not all the protection bits are programmed):

The protection condition can be canceled by the automatic protection bit erase operation. With this device, protection bits set by an individual block can be erased handling all the blocks at a time as shown in Table 23-19. The target bits are specified in the seventh bus write cycle and when the command is completed, the device is in a condition all the blocks are erased. The protection status of each block can be checked by FLCS <BLPRO> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the protection bits of FLCS <BLPRO> selected for erasure are set to "0."

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

(Note) The FLCS <RDY/BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.

6) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (recommended input data is 0x00). On and after the fourth bus write cycle, when an arbitrary flash memory area is read, the ID value will be loaded. Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and ID-Read commands can be repetitively executed. For returning to the read mode, use the Read/reset command or hardware reset command.

(5) Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the protection status of each block.

Table 23-13 Flash Control Register

FCFLCS
0x41FF_F020

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

	23	22	21	20	19	18	17	16
bit Symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
Read/Write	R		R	R	R	R	R	R
After reset	0		(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)	(Note 2)
Function	"0" is read.		Protection for Block 5 0: disabled 1:enabled	Protection for Block 4 0: disabled 1:enabled	Protection for Block 3 0: disabled 1:enabled	Protection for Block 2 0: disabled 1:enabled	Protection for Block 1 0: disabled 1:enabled	Protection for Block 0 0: disabled 1:enabled

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

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	RDY/BSY
Read/Write	R							R
After reset	0							1
Function	"0" is read.							Ready/Busy (Note 1) 0:Auto operating 1:Auto operation terminated

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Bit [21:16]: Protection status bits

Each of the protection bits represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

(Note 1) This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 microseconds regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

(Note 2) The value varies depending on protection applied.

Table 23-14 Security bit register

FCSECBIT
0x41FF_F010

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

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

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

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	SECBIT
Read/Write	R							R/W
After reset	0							1
Function	"0" is read							Security bits 0:disabled 1:enabled

(Note) This register is initialized only by power-on reset.

(6) List of Command Sequences

Table 23-15 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	–	–	–	–	–	–
	0xF0	–	–	–	–	–	–
Read/Reset	0x54XX	0xAAXX	0x54XX	RA	–	–	–
	0xAA	0x55	0xF0	RD	–	–	–
ID-Read	0x54XX	0xAAXX	0x54XX	IA	0xXX	–	–
	0xAA	0x55	0x90	0x00	ID	–	–
Automatic page programming	0x54XX	0xAAXX	0x54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Auto Block erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Protection bit programming	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x54XX	0xAAXX	0x54XX	0x54XX	0xAAXX	0x54XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

Supplementary explanation

- RA: Read address
- RD: Read data
- IA: ID address
- ID: ID data
- PA: Program page address
  PD: Program data (32 bit data)
  After the fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

(Note 1) Always set "0" to the address bits [1:0] in the entire bus cycle. (Recommendable setting values to bits [7:2] are ”0”.)
(Note 2) Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by 32-bit data transfer commands. The address [31:16] in each bus write cycle should be the target flash memory address [31:16] of the command sequence. Use "Addr." in the table for the address [15:0].

(7) Address bit configuration for bus write cycles

Table 23-16 Address Bit Configuration for Bus Write Cycles

Address	Addr [31:19]	Addr [18]	Addr [17]	Addr [16]	Addr [15]	Addr [14]	Addr [13:11]	Addr [10]	Addr [9]	Addr [8]	Addr [7:0]

Normal commands	**Normal bus write cycle address configuration**			
	Flash area	"0" is recommended.	Command	Addr[1:0]="0" (fixed) Others:0 (recommended)
ID -READ	**IA: ID address (Set the fourth bus write cycle address for ID-Read operation)**			
	Flash area	"0" is recommended.	ID address	Addr[1:0]="0" (fixed) , Others:0 (recommended)

Block erase	**BA: Block address (Set the sixth bus write cycle address for block erase operation)**				
	Block selection (Table 23-17)		Addr[1:0]="0" (fixed) , Others:0 (recommended)		
Auto page programming	**PA: Program page address (Set the fourth bus write cycle address for page programming operation)**				
	Page selection				Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit programming	**PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)**				
	Flash area	Protection bit selection (Table 23-18)	Fixed to "0".	Protection bit selection (Table 23-18)	Addr[1:0]="0" (fixed) Others:0 (recommended)
Protection bit erase	**PBA: Protection bit address (Set the seventh bus erase cycle address for protection bit erasure)**				
	Flash area	Protection bit selection (Table 23-19)	"0" is recommended.	Protection bit selection (Table 23-19)	Addr[1:0]="0" (fixed) Others:0 (recommended)
Block erase	**BA: Block address (Set the sixth bus write cycle address for block erase operation)**				
	Block selection (Table 23-17)		Addr[1:0]="0" (fixed) , Others:0 (recommended)		

(Note 1) Table 23-15"Operation Modes" **can also be used.**
(Note 2) Address setting can be performed according to the "Normal bus write cycle address configuration" from the first bus cycle.
(Note 3) "0" is recommended" can be changed as necessary.

Table 23-17 Block Address Table

Block	Address (User boot mode)	Address (Single boot mode)	Size (Kbyte)
4	0x0000_0000-0x0000_3FFF	0x3F80_0000-0x3F80_3FFF	16
5	0x0000_4000-0x0000_7FFF	0x3F80_4000-0x3F80_7FFF	16
3	0x0000_8000-0x0000_FFFF	0x3F80_8000-0x3F80_FFFF	32
2	0x0001_0000-0x0001_FFFF	0x3F81_0000-0x3F81_FFFF	64
1	0x0002_0000-0x0002_FFFF	0x3F82_0000-0x3F82_FFFF	64
0	0x0003_0000-0x0003_FFFF	0x3F83_8000-0x3F83_FFFF	64

As block address, specify any address in the block to be erased.

(Note) As for the addresses from the first to the fifth bus cycles, specify the upper 4 bit with the corresponding flash memory addresses of the blocks to be erased.

Table 23-18 Protection Bit Programming Address Table

Block	Protection bit	The seventh bus write cycle address						
		Address [18]	Address [17]	Address [16]	Address [15:11]	Address [10]	Address [9]	Address [8]
Block0	BLPRO0	0	0	Fixed to “0”.			0	0
Block1	BLPRO1	0	0				0	1
Block2	BLPRO2	0	0				1	0
Block3	BLPRO3	0	0				1	1
Block4	BLPRO4	0	1				0	0
Block5	BLPRO5	0	1				0	1

Table 23-19 Protection Bit Erase Address Table

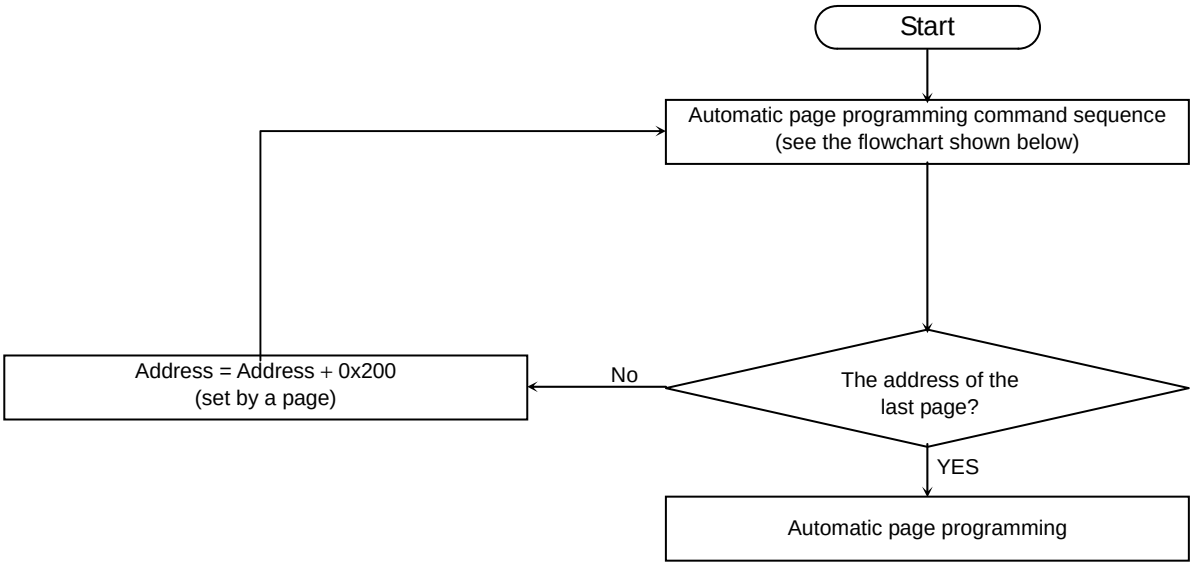
Block	Protection bit	The seventh bus write cycle address [18:17]	
		Address [18]	Address [17]
Block0 to 3	BLPRO0 to 3	0	0
Block4 to 5	BLPRO4 to 5	0	1

(Note) The protection bit erase command cannot erase by individual block.

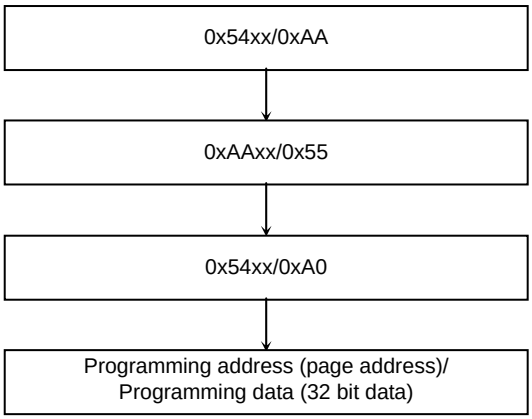
Table 23-20 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following 32-bit data transfer command (ID)

IA [15:14]	ID [7: 0]	Code
00b	0x98	Manufacturer code
01b	0x5A	Device code
10b	Reserved	---
11b	0x13	Macro code

(8) Flowchart



Automatic Page Programming Command Sequence (Address/ Command)



(Note) Command sequence is executed by 0x54xx or 0x55xx.

Fig 23-14 Automatic Programming

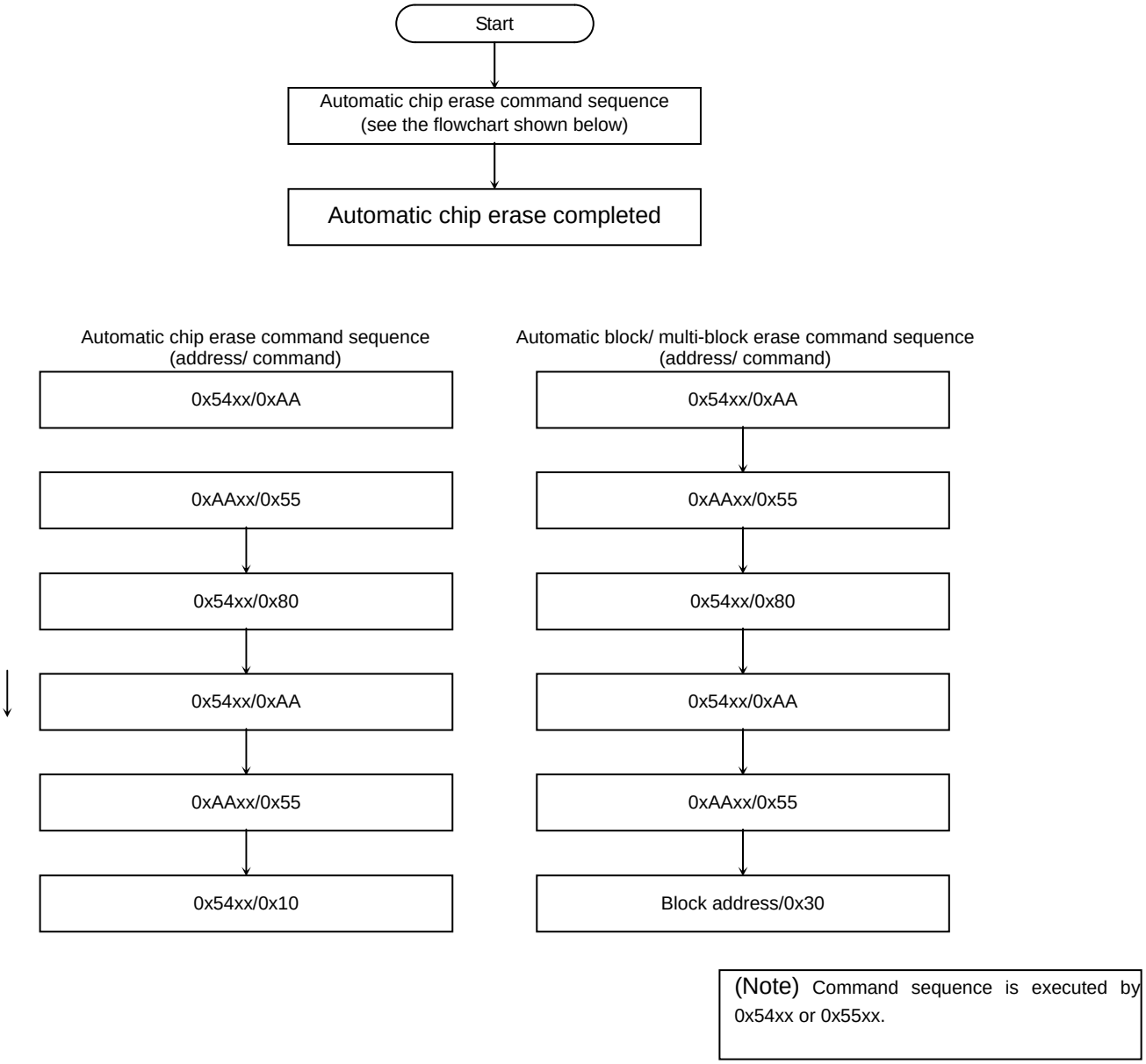


Fig 23-15 Automatic Erase

24 Protect/security function

Important
TMPM380FW ,M382FW(128K version) does not Flash memory BLOCK0,BLOCK1.
Please do not use these functions if you use this product.

TMPM382FS (64K version) does not Flash memory BLOCK0,BLOCK1,BLOCK2.
Please do not use these functions if you use this product.

24.1 OutLine

The TMPM380 offers two kinds of ROM protect/ security functions. One is a write/ erase-protect function for the internal flash ROM data. The other is a security function that restricts internal flash ROM data readout and debugging.

24.2 Feature

24.2.1 Internal Flash ROM write/erase protect

The write/ erase-protect function enables the internal flash to prohibit the writing and erasing operation for each block.

This function is available with a single chip mode, single boot mode and writer mode. To activate the function, write “1” to the corresponding bits to a block to protect. Writing “0” to the bits cancels the protection. The protection status of the bits can be monitored by the FCFLCS <BLPRO> bit.

24.2.2 Security function

The security function restricts flash ROM data readout and debugging.
This function is available under the conditions shown below.

1) The FCSECBIT <SECBIT> bit is set to”1”.
2) All the protection bits (the FCFLCS<BLPRO> bits) used to the write/erase-protect function are set to “1”.

Note) The FCSECBIT <SECBIT> bit is set to “1” at a power-on reset right after power-on.

Table 24-1 shows details of the restrictions by the security function.

Table 24-1 Restrictions by the security function

Item	Details
1) ROM data readout	Data in the ROM area cannot be read out when writer mode is set. By executing readout, the company code 0x0098 is read. The ROM reading operation is available with a single chip mode and single boot mode.
2) Debug port	Communication of JTAG/SW and trace are prohibited.
3) Command for flash memory	Writing a command to the flash memory is prohibited. An attempt to erase the contents in the bits used for the write/erase-protection will erase all the contents of flash memory include protection bits.

24.3 Resisters

The flash control register shows the status of the flash memory operation and the protection of each block.

Table 24-2 Flash control resister

FCFLCS
0x41FF_F020

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

	23	22	21	20	19	18	17	16
bit Symbol	-	-	BLPRO5	BLPRO4	BLPRO3	BLPRO2	BLPRO1	BLPRO0
Read/Write	R		R	R	R	R	R	R
after Reset	0		(note2)	(note2)	(note2)	(note2)	(note2)	(note2)
function	Reading data is "0"		Block5 protect status 0: no protect status. 1: protect status.	Block4 protect status 0: no protect status. 1: protect status.	Block3 protect status 0: no protect status. 1: protect status.	Block2 protect status 0: no protect status. 1: protect status.	Block1 protect status 0: no protect status. 1: protect status.	Block0 protect status 0: no protect status. 1: protect status.

	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	RDY/BSY
Read/Write	R							R
after reset	0							1
Function	reading data is "0"							Ready/ Busy (note1) 0:under automatic operation 1:automatic operation is finished

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided as a means to monitor the status of automatic operation. . When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

Bit [21:16]: Protection status bits

Each of the protection bits (6 bits) represents the protection status of the corresponding block. When a bit is set to "1", it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

(Note 1) This command must be issued in the ready state. Issuing the command in the busy state may disable both correct command transmission and further command input. To exit from the condition, execute system reset. System reset requires at least 0.5 microseconds regardless of the system clock frequency. In this condition, it takes approx. 2 ms to enable reading after reset.

(Note 2) The value varies depending on protection status.

Table 24-3 security bit resister

FCSECBIT
0x41FF_F010

	31	30	29	28	27	26	25	24
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	23	22	21	20	19	18	17	16
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	15	14	13	12	11	10	9	8
bit Symbol	-	-	-	-	-	-	-	-
Read/Write	R							
after reset	0							
function	reading data is "0"							

	7	6	5	4	3	2	1	0
bit Symbol	-	-	-	-	-	-	-	SECBIT
Read/Write	R							R/W
after reset	0							1
function	reading data is "0"							security bit 0: disable 1: enable

(Note) This register is initialized only by power-on reset.

24.4 Writing and erasing

24.4.1 Protection bits

Writing and erasing protection bits are available with a single chip mode, single boot mode and writer mode.
Writing to the protection bits is done on block-by-block basis.
Erasing of the protection bits is done by two groups of the blocks: block 0 through 3 and block 4 through 5. When the settings for all the blocks are “1”, erasing must be done after clearing the FCSECBIT <SECBIT> bit to “0”. An attempt to erase protection bits when <SECBIT> bit is “1”, it will erases all the contents of flash memory include protection bits. To write and erase the protection bits, command sequence is used.
See chapter 23 Flash Memory Operation for details.

24.4.2 Security bit

The FCSECBIT <SECBIT> bit that activates security function is set to “1” at a power-on reset right after power-on. It can be rewritten at single chip mode and single boot mode.by the following procedure.

1) Write the code 0xa74a9d23 to FCSECBIT register.
2) Write data within 16 clocks from the writing above.

Note) The above procedure is enabled only when using 32-bit data transfer command.

25 Special Function Registers

[1] Port registers
[2] 16-bit timer (TMRB)
[3] Encoder input (ENC)
[4] Serial bus interface (SBI)
[5] Serial interface (UART/SIO)
[6] 12-bit A/D converter (A/DC)
[7] Watchdog timer (WDT)
[8] Real time clock (RTC)
[9] Clock generator (CG)
[10] Remote control signal preprocessor (RMC)
[11] Oscillation frequency detector (OFD)
[12] Power on reset (POR), Voltage detecting circuit (VLTD)
[13] Multi purpose timer (MPT (TMR,IGBT,PMD))
[14] DMA controller (DMAC)
[15] SSP controller
[16] Flash controller

(Note 1)	**As for the internal I/O areas (0x4000_0000~0x4007_FFFF), reading the areas not described in this chapter yields undefined value. Writing these areas is ignored.**
(Note 2)	**<R0> means 0(zero) is read. Writing data is disregarded.**
(Note3)	**Access to the <Reserved> areas is prohibited.**

25.1 Addresses for TMPM380

25.1.1 [1] Port [1/5]

<PORT A>

Address	Register name
0x4000_0000	PADATA
0x4000_0001	<R0>
0x4000_0002	<R0>
0x4000_0003	<R0>
0x4000_0004	PACR
0x4000_0005	<R0>
0x4000_0006	<R0>
0x4000_0007	<R0>
0x4000_0008	PAFR1
0x4000_0009	<R0>
0x4000_000A	<R0>
0x4000_000B	<R0>
0x4000_000C	PAFR2
0x4000_000D	<R0>
0x4000_000E	<R0>
0x4000_000F	<R0>

Address	Register name
0x4000_0010	
0x4000_0011	
0x4000_0012	
0x4000_0013	
0x4000_0014	
0x4000_0015	
0x4000_0016	
0x4000_0017	
0x4000_0018	
0x4000_0019	
0x4000_001A	
0x4000_001B	
0x4000_001C	
0x4000_001D	
0x4000_001E	
0x4000_001F	

Address	Register name
0x4000_0020	
0x4000_0021	
0x4000_0022	
0x4000_0023	
0x4000_0024	
0x4000_0025	
0x4000_0026	
0x4000_0027	
0x4000_0028	PAOD
0x4000_0029	<R0>
0x4000_002A	<R0>
0x4000_002B	<R0>
0x4000_002C	PAPUP
0x4000_002D	<R0>
0x4000_002E	<R0>
0x4000_002F	<R0>

Address	Register name
0x4000_0030	PAPDN
0x4000_0031	<R0>
0x4000_0032	<R0>
0x4000_0033	<R0>
0x4000_0034	
0x4000_0035	
0x4000_0036	
0x4000_0037	
0x4000_0038	PAIE
0x4000_0039	<R0>
0x4000_003A	<R0>
0x4000_003B	<R0>
0x4000_003C	
0x4000_003D	
0x4000_003E	
0x4000_003F	

<PORT B>

Address	Register name
0x4000_0040	PBDATA
0x4000_0041	<R0>
0x4000_0042	<R0>
0x4000_0043	<R0>
0x4000_0044	PBCR
0x4000_0045	<R0>
0x4000_0046	<R0>
0x4000_0047	<R0>
0x4000_0048	PBFR1
0x4000_0049	<R0>
0x4000_004A	<R0>
0x4000_004B	<R0>
0x4000_004C	
0x4000_004D	
0x4000_004E	
0x4000_004F	

Address	Register name
0x4000_0050	
0x4000_0051	
0x4000_0052	
0x4000_0053	
0x4000_0054	
0x4000_0055	
0x4000_0056	
0x4000_0057	
0x4000_0058	
0x4000_0059	
0x4000_005A	
0x4000_005B	
0x4000_005C	
0x4000_005D	
0x4000_005E	
0x4000_005F	

Address	Register name
0x4000_0060	
0x4000_0061	
0x4000_0062	
0x4000_0063	
0x4000_0064	
0x4000_0065	
0x4000_0066	
0x4000_0067	
0x4000_0068	PBOD
0x4000_0069	<R0>
0x4000_006A	<R0>
0x4000_006B	<R0>
0x4000_006C	PBPUP
0x4000_006D	<R0>
0x4000_006E	<R0>
0x4000_006F	<R0>

Address	Register name
0x4000_0070	PBPDN
0x4000_0071	<R0>
0x4000_0072	<R0>
0x4000_0073	<R0>
0x4000_0074	
0x4000_0075	
0x4000_0076	
0x4000_0077	
0x4000_0078	PBIE
0x4000_0079	<R0>
0x4000_007A	<R0>
0x4000_007B	<R0>
0x4000_007C	
0x4000_007D	
0x4000_007E	
0x4000_007F	

<PORT C>

Address	Register name
0x4000_0080	PCDATA
0x4000_0081	<R0>
0x4000_0082	<R0>
0x4000_0083	<R0>
0x4000_0084	PCCR
0x4000_0085	<R0>
0x4000_0086	<R0>
0x4000_0087	<R0>
0x4000_0088	PCFR1
0x4000_0089	<R0>
0x4000_008A	<R0>
0x4000_008B	<R0>
0x4000_008C	PCFR2
0x4000_008D	<R0>
0x4000_008E	<R0>
0x4000_008F	<R0>

Address	Register name
0x4000_0090	PCFR3
0x4000_0091	<R0>
0x4000_0092	<R0>
0x4000_0093	<R0>
0x4000_0094	PCFR4
0x4000_0095	<R0>
0x4000_0096	<R0>
0x4000_0097	<R0>
0x4000_0098	PCFR5
0x4000_0099	<R0>
0x4000_009A	<R0>
0x4000_009B	<R0>
0x4000_009C	
0x4000_009D	
0x4000_009E	
0x4000_009F	

Address	Register name
0x4000_00A0	
0x4000_00A1	
0x4000_00A2	
0x4000_00A3	
0x4000_00A4	
0x4000_00A5	
0x4000_00A6	
0x4000_00A7	
0x4000_00A8	PCOD
0x4000_00A9	<R0>
0x4000_00AA	<R0>
0x4000_00AB	<R0>
0x4000_00AC	PCPUP
0x4000_00AD	<R0>
0x4000_00AE	<R0>
0x4000_00AF	<R0>

Address	Register name
0x4000_00B0	PCPDN
0x4000_00B1	<R0>
0x4000_00B2	<R0>
0x4000_00B3	<R0>
0x4000_00B4	
0x4000_00B5	
0x4000_00B6	
0x4000_00B7	
0x4000_00B8	PCIE
0x4000_00B9	<R0>
0x4000_00BA	<R0>
0x4000_00BB	<R0>
0x4000_00BC	
0x4000_00BD	
0x4000_00BE	
0x4000_00BF	

[1] Port [2/5]

<PORT D>

Address	Register name
0x4000_00C0	PDDATA
0x4000_00C1	<R0>
0x4000_00C2	<R0>
0x4000_00C3	<R0>
0x4000_00C4	PDCR
0x4000_00C5	<R0>
0x4000_00C6	<R0>
0x4000_00C7	<R0>
0x4000_00C8	PDFR1
0x4000_00C9	<R0>
0x4000_00CA	<R0>
0x4000_00CB	<R0>
0x4000_00CC	PDFR2
0x4000_00CD	<R0>
0x4000_00CE	<R0>
0x4000_00CF	<R0>

Address	Register name
0x4000_00D0	PDFR3
0x4000_00D1	<R0>
0x4000_00D2	<R0>
0x4000_00D3	<R0>
0x4000_00D4	
0x4000_00D5	
0x4000_00D6	
0x4000_00D7	
0x4000_00D8	
0x4000_00D9	
0x4000_00DA	
0x4000_00DB	
0x4000_00DC	
0x4000_00DD	
0x4000_00DE	
0x4000_00DF	

Address	Register name
0x4000_00E0	
0x4000_00E1	
0x4000_00E2	
0x4000_00E3	
0x4000_00E4	
0x4000_00E5	
0x4000_00E6	
0x4000_00E7	
0x4000_00E8	PDOD
0x4000_00E9	<R0>
0x4000_00EA	<R0>
0x4000_00EB	<R0>
0x4000_00EC	PDPUP
0x4000_00ED	<R0>
0x4000_00EE	<R0>
0x4000_00EF	<R0>

Address	Register name
0x4000_00F0	PDPDN
0x4000_00F1	<R0>
0x4000_00F2	<R0>
0x4000_00F3	<R0>
0x4000_00F4	
0x4000_00F5	
0x4000_00F6	
0x4000_00F7	
0x4000_00F8	PDIE
0x4000_00F9	<R0>
0x4000_00FA	<R0>
0x4000_00FB	<R0>
0x4000_00FC	
0x4000_00FD	
0x4000_00FE	
0x4000_00FF	

<PORT E>

Address	Register name
0x4000_0100	PEDATA
0x4000_0101	<R0>
0x4000_0102	<R0>
0x4000_0103	<R0>
0x4000_0104	PECR
0x4000_0105	<R0>
0x4000_0106	<R0>
0x4000_0107	<R0>
0x4000_0108	PEFR1
0x4000_0109	<R0>
0x4000_010A	<R0>
0x4000_010B	<R0>
0x4000_010C	PEFR2
0x4000_010D	<R0>
0x4000_010E	<R0>
0x4000_010F	<R0>

Address	Register name
0x4000_0110	
0x4000_0111	
0x4000_0112	
0x4000_0113	
0x4000_0114	
0x4000_0115	
0x4000_0116	
0x4000_0117	
0x4000_0118	
0x4000_0119	
0x4000_011A	
0x4000_011B	
0x4000_011C	
0x4000_011D	
0x4000_011E	
0x4000_011F	

Address	Register name
0x4000_0120	
0x4000_0121	
0x4000_0122	
0x4000_0123	
0x4000_0124	
0x4000_0125	
0x4000_0126	
0x4000_0127	
0x4000_0128	PEOD
0x4000_0129	<R0>
0x4000_012A	<R0>
0x4000_012B	<R0>
0x4000_012C	PEPUP
0x4000_012D	<R0>
0x4000_012E	<R0>
0x4000_012F	<R0>

Address	Register name
0x4000_0130	PEPDN
0x4000_0131	<R0>
0x4000_0132	<R0>
0x4000_0133	<R0>
0x4000_0134	
0x4000_0135	
0x4000_0136	
0x4000_0137	
0x4000_0138	PEIE
0x4000_0139	<R0>
0x4000_013A	<R0>
0x4000_013B	<R0>
0x4000_013C	
0x4000_013D	
0x4000_013E	
0x4000_013F	

<PORT F>

Address	Register name
0x4000_0140	PFDATA
0x4000_0141	<R0>
0x4000_0142	<R0>
0x4000_0143	<R0>
0x4000_0144	PFCR
0x4000_0145	<R0>
0x4000_0146	<R0>
0x4000_0147	<R0>
0x4000_0148	PFFR1
0x4000_0149	<R0>
0x4000_014A	<R0>
0x4000_014B	<R0>
0x4000_014C	PFFR2
0x4000_014D	<R0>
0x4000_014E	<R0>
0x4000_014F	<R0>

Address	Register name
0x4000_0150	PFFR3
0x4000_0151	<R0>
0x4000_0152	<R0>
0x4000_0153	<R0>
0x4000_0154	
0x4000_0155	
0x4000_0156	
0x4000_0157	
0x4000_0158	
0x4000_0159	
0x4000_015A	
0x4000_015B	
0x4000_015C	
0x4000_015D	
0x4000_015E	
0x4000_015F	

Address	Register name
0x4000_0160	
0x4000_0161	
0x4000_0162	
0x4000_0163	
0x4000_0164	
0x4000_0165	
0x4000_0166	
0x4000_0167	
0x4000_0168	PFOD
0x4000_0169	<R0>
0x4000_016A	<R0>
0x4000_016B	<R0>
0x4000_016C	PFPUP
0x4000_016D	<R0>
0x4000_016E	<R0>
0x4000_016F	<R0>

Address	Register name
0x4000_0170	PFPDN
0x4000_0171	<R0>
0x4000_0172	<R0>
0x4000_0173	<R0>
0x4000_0174	
0x4000_0175	
0x4000_0176	
0x4000_0177	
0x4000_0178	PFIE
0x4000_0179	<R0>
0x4000_017A	<R0>
0x4000_017B	<R0>
0x4000_017C	
0x4000_017D	
0x4000_017E	
0x4000_017F	

[1] Port [3/5]

<PORT G>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4000_0180	PGDATA	0x4000_0190	PGFR3	0x4000_01A0		0x4000_01B0	PGPDN
0x4000_0181	<R0>	0x4000_0191	<R0>	0x4000_01A1		0x4000_01B1	<R0>
0x4000_0182	<R0>	0x4000_0192	<R0>	0x4000_01A2		0x4000_01B2	<R0>
0x4000_0183	<R0>	0x4000_0193	<R0>	0x4000_01A3		0x4000_01B3	<R0>
0x4000_0184	PGCR	0x4000_0194		0x4000_01A4		0x4000_01B4	
0x4000_0185	<R0>	0x4000_0195		0x4000_01A5		0x4000_01B5	
0x4000_0186	<R0>	0x4000_0196		0x4000_01A6		0x4000_01B6	
0x4000_0187	<R0>	0x4000_0197		0x4000_01A7		0x4000_01B7	
0x4000_0188	PGFR1	0x4000_0198		0x4000_01A8	PGOD	0x4000_01B8	PGIE
0x4000_0189	<R0>	0x4000_0199		0x4000_01A9	<R0>	0x4000_01B9	<R0>
0x4000_018A	<R0>	0x4000_019A		0x4000_01AA	<R0>	0x4000_01BA	<R0>
0x4000_018B	<R0>	0x4000_019B		0x4000_01AB	<R0>	0x4000_01BB	<R0>
0x4000_018C	PGFR2	0x4000_019C		0x4000_01AC	PGPUP	0x4000_01BC	
0x4000_018D	<R0>	0x4000_019D		0x4000_01AD	<R0>	0x4000_01BD	
0x4000_018E	<R0>	0x4000_019E		0x4000_01AE	<R0>	0x4000_01BE	
0x4000_018F	<R0>	0x4000_019F		0x4000_01AF	<R0>	0x4000_01BF	

<PORT H>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4000_01C0	PHDATA	0x4000_01D0		0x4000_01E0		0x4000_01F0	PHPDN
0x4000_01C1	<R0>	0x4000_01D1		0x4000_01E1		0x4000_01F1	<R0>
0x4000_01C2	<R0>	0x4000_01D2		0x4000_01E2		0x4000_01F2	<R0>
0x4000_01C3	<R0>	0x4000_01D3		0x4000_01E3		0x4000_01F3	<R0>
0x4000_01C4	PHCR	0x4000_01D4		0x4000_01E4		0x4000_01F4	
0x4000_01C5	<R0>	0x4000_01D5		0x4000_01E5		0x4000_01F5	
0x4000_01C6	<R0>	0x4000_01D6		0x4000_01E6		0x4000_01F6	
0x4000_01C7	<R0>	0x4000_01D7		0x4000_01E7		0x4000_01F7	
0x4000_01C8	PHFR1	0x4000_01D8		0x4000_01E8	PHOD	0x4000_01F8	PHIE
0x4000_01C9	<R0>	0x4000_01D9		0x4000_01E9	<R0>	0x4000_01F9	<R0>
0x4000_01CA	<R0>	0x4000_01DA		0x4000_01EA	<R0>	0x4000_01FA	<R0>
0x4000_01CB	<R0>	0x4000_01DB		0x4000_01EB	<R0>	0x4000_01FB	<R0>
0x4000_01CC		0x4000_01DC		0x4000_01EC	PHPUP	0x4000_01FC	
0x4000_01CD		0x4000_01DD		0x4000_01ED	<R0>	0x4000_01FD	
0x4000_01CE		0x4000_01DE		0x4000_01EE	<R0>	0x4000_01FE	
0x4000_01CF		0x4000_01DF		0x4000_01EF	<R0>	0x4000_01FF	

<PORT I>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4000_0200	PIDATA	0x4000_0210		0x4000_0220		0x4000_0230	PIPDN
0x4000_0201	<R0>	0x4000_0211		0x4000_0221		0x4000_0231	<R0>
0x4000_0202	<R0>	0x4000_0212		0x4000_0222		0x4000_0232	<R0>
0x4000_0203	<R0>	0x4000_0213		0x4000_0223		0x4000_0233	<R0>
0x4000_0204	PICR	0x4000_0214		0x4000_0224		0x4000_0234	
0x4000_0205	<R0>	0x4000_0215		0x4000_0225		0x4000_0235	
0x4000_0206	<R0>	0x4000_0216		0x4000_0226		0x4000_0236	
0x4000_0207	<R0>	0x4000_0217		0x4000_0227		0x4000_0237	
0x4000_0208		0x4000_0218		0x4000_0228	PIOD	0x4000_0238	PIIE
0x4000_0209		0x4000_0219		0x4000_0229	<R0>	0x4000_0239	<R0>
0x4000_020A		0x4000_021A		0x4000_022A	<R0>	0x4000_023A	<R0>
0x4000_020B		0x4000_021B		0x4000_022B	<R0>	0x4000_023B	<R0>
0x4000_020C		0x4000_021C		0x4000_022C	PIPUP	0x4000_023C	
0x4000_020D		0x4000_021D		0x4000_022D	<R0>	0x4000_023D	
0x4000_020E		0x4000_021E		0x4000_022E	<R0>	0x4000_023E	
0x4000_020F		0x4000_021F		0x4000_022F	<R0>	0x4000_023F	

[1] Port [4/5]

<PORT J>

Address	Register name
0x4000_0240	PJDATA
0x4000_0241	<R0>
0x4000_0242	<R0>
0x4000_0243	<R0>
0x4000_0244	PJCR
0x4000_0245	<R0>
0x4000_0246	<R0>
0x4000_0247	<R0>
0x4000_0248	PJFR1
0x4000_0249	<R0>
0x4000_024A	<R0>
0x4000_024B	<R0>
0x4000_024C	
0x4000_024D	
0x4000_024E	
0x4000_024F	

Address	Register name
0x4000_0250	
0x4000_0251	
0x4000_0252	
0x4000_0253	
0x4000_0254	
0x4000_0255	
0x4000_0256	
0x4000_0257	
0x4000_0258	
0x4000_0259	
0x4000_025A	
0x4000_025B	
0x4000_025C	
0x4000_025D	
0x4000_025E	
0x4000_025F	

Address	Register name
0x4000_0260	
0x4000_0261	
0x4000_0262	
0x4000_0263	
0x4000_0264	
0x4000_0265	
0x4000_0266	
0x4000_0267	
0x4000_0268	PJOD
0x4000_0269	<R0>
0x4000_026A	<R0>
0x4000_026B	<R0>
0x4000_026C	PJPUP
0x4000_026D	<R0>
0x4000_026E	<R0>
0x4000_026F	<R0>

Address	Register name
0x4000_0270	PJPDN
0x4000_0271	<R0>
0x4000_0272	<R0>
0x4000_0273	<R0>
0x4000_0274	
0x4000_0275	
0x4000_0276	
0x4000_0277	
0x4000_0278	PJIE
0x4000_0279	<R0>
0x4000_027A	<R0>
0x4000_027B	<R0>
0x4000_027C	
0x4000_027D	
0x4000_027E	
0x4000_027F	

Address	Register name
0x4000_0280	
0x4000_0281	
0x4000_0282	
0x4000_0283	
0x4000_0284	
0x4000_0285	
0x4000_0286	
0x4000_0287	
0x4000_0288	
0x4000_0289	
0x4000_028A	
0x4000_028B	
0x4000_028C	
0x4000_028D	
0x4000_028E	
0x4000_028F	

Address	Register name
0x4000_0290	
0x4000_0291	
0x4000_0292	
0x4000_0293	
0x4000_0294	
0x4000_0295	
0x4000_0296	
0x4000_0297	
0x4000_0298	
0x4000_0299	
0x4000_029A	
0x4000_029B	
0x4000_029C	
0x4000_029D	
0x4000_029E	
0x4000_029F	

Address	Register name
0x4000_02A0	
0x4000_02A1	
0x4000_02A2	
0x4000_02A3	
0x4000_02A4	
0x4000_02A5	
0x4000_02A6	
0x4000_02A7	
0x4000_02A8	
0x4000_02A9	
0x4000_02AA	
0x4000_02AB	
0x4000_02AC	
0x4000_02AD	
0x4000_02AE	
0x4000_02AF	

Address	Register name
0x4000_02B0	
0x4000_02B1	
0x4000_02B2	
0x4000_02B3	
0x4000_02B4	
0x4000_02B5	
0x4000_02B6	
0x4000_02B7	
0x4000_02B8	
0x4000_02B9	
0x4000_02BA	
0x4000_02BB	
0x4000_02BC	
0x4000_02BD	
0x4000_02BE	
0x4000_02BF	

<PORT L>

Address	Register name
0x4000_02C0	PLDATA
0x4000_02C1	<R0>
0x4000_02C2	<R0>
0x4000_02C3	<R0>
0x4000_02C4	PLCR
0x4000_02C5	<R0>
0x4000_02C6	<R0>
0x4000_02C7	<R0>
0x4000_02C8	PLFR1
0x4000_02C9	<R0>
0x4000_02CA	<R0>
0x4000_02CB	<R0>
0x4000_02CC	
0x4000_02CD	
0x4000_02CE	
0x4000_02CF	

Address	Register name
0x4000_02D0	
0x4000_02D1	
0x4000_02D2	
0x4000_02D3	
0x4000_02D4	
0x4000_02D5	
0x4000_02D6	
0x4000_02D7	
0x4000_02D8	
0x4000_02D9	
0x4000_02DA	
0x4000_02DB	
0x4000_02DC	
0x4000_02DD	
0x4000_02DE	
0x4000_02DF	

Address	Register name
0x4000_02E0	
0x4000_02E1	
0x4000_02E2	
0x4000_02E3	
0x4000_02E4	
0x4000_02E5	
0x4000_02E6	
0x4000_02E7	
0x4000_02E8	PLOD
0x4000_02E9	<R0>
0x4000_02EA	<R0>
0x4000_02EB	<R0>
0x4000_02EC	PLPUP
0x4000_02ED	<R0>
0x4000_02EE	<R0>
0x4000_02EF	<R0>

Address	Register name
0x4000_02F0	PLPDN
0x4000_02F1	<R0>
0x4000_02F2	<R0>
0x4000_02F3	<R0>
0x4000_02F4	
0x4000_02F5	
0x4000_02F6	
0x4000_02F7	
0x4000_02F8	PLIE
0x4000_02F9	<R0>
0x4000_02FA	<R0>
0x4000_02FB	<R0>
0x4000_02FC	
0x4000_02FD	
0x4000_02FE	
0x4000_02FF	

[1] Port [5/5]

<PORT M>

Address	Register name
0x4000_0300	PMDATA
0x4000_0301	<R0>
0x4000_0302	<R0>
0x4000_0303	<R0>
0x4000_0304	PMCR
0x4000_0305	<R0>
0x4000_0306	<R0>
0x4000_0307	<R0>
0x4000_0308	
0x4000_0309	
0x4000_030A	
0x4000_030B	
0x4000_030C	
0x4000_030D	
0x4000_030E	
0x4000_030F	

Address	Register name
0x4000_0310	
0x4000_0311	
0x4000_0312	
0x4000_0313	
0x4000_0314	
0x4000_0315	
0x4000_0316	
0x4000_0317	
0x4000_0318	
0x4000_0319	
0x4000_031A	
0x4000_031B	
0x4000_031C	
0x4000_031D	
0x4000_031E	
0x4000_031F	

Address	Register name
0x4000_0320	
0x4000_0321	
0x4000_0322	
0x4000_0323	
0x4000_0324	
0x4000_0325	
0x4000_0326	
0x4000_0327	
0x4000_0328	PMOD
0x4000_0329	<R0>
0x4000_032A	<R0>
0x4000_032B	<R0>
0x4000_032C	PMPUP
0x4000_032D	<R0>
0x4000_032E	<R0>
0x4000_032F	<R0>

Address	Register name
0x4000_0330	PMPDN
0x4000_0331	<R0>
0x4000_0332	<R0>
0x4000_0333	<R0>
0x4000_0334	
0x4000_0335	
0x4000_0336	
0x4000_0337	
0x4000_0338	PMIE
0x4000_0339	<R0>
0x4000_033A	<R0>
0x4000_033B	<R0>
0x4000_033C	
0x4000_033D	
0x4000_033E	
0x4000_033F	

<PORT N>

Address	Register name
0x4000_0340	PNDATA
0x4000_0341	<R0>
0x4000_0342	<R0>
0x4000_0343	<R0>
0x4000_0344	PNCR
0x4000_0345	<R0>
0x4000_0346	<R0>
0x4000_0347	<R0>
0x4000_0348	PNFR1
0x4000_0349	<R0>
0x4000_034A	<R0>
0x4000_034B	<R0>
0x4000_034C	PNFR2
0x4000_034D	<R0>
0x4000_034E	<R0>
0x4000_034F	<R0>

Address	Register name
0x4000_0350	
0x4000_0351	
0x4000_0352	
0x4000_0353	
0x4000_0354	
0x4000_0355	
0x4000_0356	
0x4000_0357	
0x4000_0358	
0x4000_0359	
0x4000_035A	
0x4000_035B	
0x4000_035C	
0x4000_035D	
0x4000_035E	
0x4000_035F	

Address	Register name
0x4000_0360	
0x4000_0361	
0x4000_0362	
0x4000_0363	
0x4000_0364	
0x4000_0365	
0x4000_0366	
0x4000_0367	
0x4000_0368	PNOD
0x4000_0369	<R0>
0x4000_036A	<R0>
0x4000_036B	<R0>
0x4000_036C	PNPUP
0x4000_036D	<R0>
0x4000_036E	<R0>
0x4000_036F	<R0>

Address	Register name
0x4000_0370	PNPDN
0x4000_0371	<R0>
0x4000_0372	<R0>
0x4000_0373	<R0>
0x4000_0374	
0x4000_0375	
0x4000_0376	
0x4000_0377	
0x4000_0378	PNIE
0x4000_0379	<R0>
0x4000_037A	<R0>
0x4000_037B	<R0>
0x4000_037C	
0x4000_037D	
0x4000_037E	
0x4000_037F	

<PORT P>

Address	Register name
0x4000_0380	PPDATA
0x4000_0381	<R0>
0x4000_0382	<R0>
0x4000_0383	<R0>
0x4000_0384	PPCR
0x4000_0385	<R0>
0x4000_0386	<R0>
0x4000_0387	<R0>
0x4000_0388	
0x4000_0389	
0x4000_038A	
0x4000_038B	
0x4000_038C	
0x4000_038D	
0x4000_038E	
0x4000_038F	

Address	Register name
0x4000_0390	
0x4000_0391	
0x4000_0392	
0x4000_0393	
0x4000_0394	
0x4000_0395	
0x4000_0396	
0x4000_0397	
0x4000_0398	
0x4000_0399	
0x4000_039A	
0x4000_039B	
0x4000_039C	
0x4000_039D	
0x4000_039E	
0x4000_039F	

Address	Register name
0x4000_03A0	
0x4000_03A1	
0x4000_03A2	
0x4000_03A3	
0x4000_03A4	
0x4000_03A5	
0x4000_03A6	
0x4000_03A7	
0x4000_03A8	PPOD
0x4000_03A9	<R0>
0x4000_03AA	<R0>
0x4000_03AB	<R0>
0x4000_03AC	PPPUP
0x4000_03AD	<R0>
0x4000_03AE	<R0>
0x4000_03AF	<R0>

Address	Register name
0x4000_03B0	PPPDN
0x4000_03B1	<R0>
0x4000_03B2	<R0>
0x4000_03B3	<R0>
0x4000_03B4	
0x4000_03B5	
0x4000_03B6	
0x4000_03B7	
0x4000_03B8	PPIE
0x4000_03B9	<R0>
0x4000_03BA	<R0>
0x4000_03BB	<R0>
0x4000_03BC	
0x4000_03BD	
0x4000_03BE	
0x4000_03BF	

25.1.2 [2] 16-bit timer [1/3]

<TMRB0>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0000	TB0EN	0x4001_0010	TB0FFCR	0x4001_0020	TB0RG0	0x4001_0030	
0x4001_0001	<R0>	0x4001_0011	<R0>	0x4001_0021		0x4001_0031	
0x4001_0002	<R0>	0x4001_0012	<R0>	0x4001_0022	<R0>	0x4001_0032	
0x4001_0003	<R0>	0x4001_0013	<R0>	0x4001_0023	<R0>	0x4001_0033	
0x4001_0004	TB0RUN	0x4001_0014	TB0ST	0x4001_0024	TB0RG1	0x4001_0034	
0x4001_0005	<R0>	0x4001_0015	<R0>	0x4001_0025		0x4001_0035	
0x4001_0006	<R0>	0x4001_0016	<R0>	0x4001_0026	<R0>	0x4001_0036	
0x4001_0007	<R0>	0x4001_0017	<R0>	0x4001_0027	<R0>	0x4001_0037	
0x4001_0008	TB0CR	0x4001_0018	TB0IM	0x4001_0028	TB0CP0	0x4001_0038	
0x4001_0009	<R0>	0x4001_0019	<R0>	0x4001_0029		0x4001_0039	
0x4001_000A	<R0>	0x4001_001A	<R0>	0x4001_002A	<R0>	0x4001_003A	
0x4001_000B	<R0>	0x4001_001B	<R0>	0x4001_002B	<R0>	0x4001_003B	
0x4001_000C	TB0MOD	0x4001_001C	TB0UC	0x4001_002C	TB0CP1	0x4001_003C	
0x4001_000D	<R0>	0x4001_001D		0x4001_002D		0x4001_003D	
0x4001_000E	<R0>	0x4001_001E	<R0>	0x4001_002E	<R0>	0x4001_003E	
0x4001_000F	<R0>	0x4001_001F	<R0>	0x4001_002F	<R0>	0x4001_003F	

<TMRB1>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0040	TB1EN	0x4001_0050	TB1FFCR	0x4001_0060	TB1RG0	0x4001_0070	
0x4001_0041	<R0>	0x4001_0051	<R0>	0x4001_0061		0x4001_0071	
0x4001_0042	<R0>	0x4001_0052	<R0>	0x4001_0062	<R0>	0x4001_0072	
0x4001_0043	<R0>	0x4001_0053	<R0>	0x4001_0063	<R0>	0x4001_0073	
0x4001_0044	TB1RUN	0x4001_0054	TB1ST	0x4001_0064	TB1RG1	0x4001_0074	
0x4001_0045	<R0>	0x4001_0055	<R0>	0x4001_0065		0x4001_0075	
0x4001_0046	<R0>	0x4001_0056	<R0>	0x4001_0066	<R0>	0x4001_0076	
0x4001_0047	<R0>	0x4001_0057	<R0>	0x4001_0067	<R0>	0x4001_0077	
0x4001_0048	TB1CR	0x4001_0058	TB1IM	0x4001_0068	TB1CP0	0x4001_0078	
0x4001_0049	<R0>	0x4001_0059	<R0>	0x4001_0069		0x4001_0079	
0x4001_004A	<R0>	0x4001_005A	<R0>	0x4001_006A	<R0>	0x4001_007A	
0x4001_004B	<R0>	0x4001_005B	<R0>	0x4001_006B	<R0>	0x4001_007B	
0x4001_004C	TB1MOD	0x4001_005C	TB1UC	0x4001_006C	TB1CP1	0x4001_007C	
0x4001_004D	<R0>	0x4001_005D		0x4001_006D		0x4001_007D	
0x4001_004E	<R0>	0x4001_005E	<R0>	0x4001_006E	<R0>	0x4001_007E	
0x4001_004F	<R0>	0x4001_005F	<R0>	0x4001_006F	<R0>	0x4001_007F	

<TMRB2>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0080	TB2EN	0x4001_0090	TB2FFCR	0x4001_00A0	TB2RG0	0x4001_00B0	
0x4001_0081	<R0>	0x4001_0091	<R0>	0x4001_00A1		0x4001_00B1	
0x4001_0082	<R0>	0x4001_0092	<R0>	0x4001_00A2	<R0>	0x4001_00B2	
0x4001_0083	<R0>	0x4001_0093	<R0>	0x4001_00A3	<R0>	0x4001_00B3	
0x4001_0084	TB2RUN	0x4001_0094	TB2ST	0x4001_00A4	TB2RG1	0x4001_00B4	
0x4001_0085	<R0>	0x4001_0095	<R0>	0x4001_00A5		0x4001_00B5	
0x4001_0086	<R0>	0x4001_0096	<R0>	0x4001_00A6	<R0>	0x4001_00B6	
0x4001_0087	<R0>	0x4001_0097	<R0>	0x4001_00A7	<R0>	0x4001_00B7	
0x4001_0088	TB2CR	0x4001_0098	TB2IM	0x4001_00A8	TB2CP0	0x4001_00B8	
0x4001_0089	<R0>	0x4001_0099	<R0>	0x4001_00A9		0x4001_00B9	
0x4001_008A	<R0>	0x4001_009A	<R0>	0x4001_00AA	<R0>	0x4001_00BA	
0x4001_008B	<R0>	0x4001_009B	<R0>	0x4001_00AB	<R0>	0x4001_00BB	
0x4001_008C	TB2MOD	0x4001_009C	TB2UC	0x4001_00AC	TB2CP1	0x4001_00BC	
0x4001_008D	<R0>	0x4001_009D		0x4001_00AD		0x4001_00BD	
0x4001_008E	<R0>	0x4001_009E	<R0>	0x4001_00AE	<R0>	0x4001_00BE	
0x4001_008F	<R0>	0x4001_009F	<R0>	0x4001_00AF	<R0>	0x4001_00BF	

[2] 16-bit timer [2/3]

<TMRB3>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_00C0	TB3EN	0x4001_00D0	TB3FFCR	0x4001_00E0	TB3RG0	0x4001_00F0	
0x4001_00C1	<R0>	0x4001_00D1	<R0>	0x4001_00E1		0x4001_00F1	
0x4001_00C2	<R0>	0x4001_00D2	<R0>	0x4001_00E2	<R0>	0x4001_00F2	
0x4001_00C3	<R0>	0x4001_00D3	<R0>	0x4001_00E3	<R0>	0x4001_00F3	
0x4001_00C4	TB3RUN	0x4001_00D4	TB3ST	0x4001_00E4	TB3RG1	0x4001_00F4	
0x4001_00C5	<R0>	0x4001_00D5	<R0>	0x4001_00E5		0x4001_00F5	
0x4001_00C6	<R0>	0x4001_00D6	<R0>	0x4001_00E6	<R0>	0x4001_00F6	
0x4001_00C7	<R0>	0x4001_00D7	<R0>	0x4001_00E7	<R0>	0x4001_00F7	
0x4001_00C8	TB3CR	0x4001_00D8	TB3IM	0x4001_00E8	TB3CP0	0x4001_00F8	
0x4001_00C9	<R0>	0x4001_00D9	<R0>	0x4001_00E9		0x4001_00F9	
0x4001_00CA	<R0>	0x4001_00DA	<R0>	0x4001_00EA	<R0>	0x4001_00FA	
0x4001_00CB	<R0>	0x4001_00DB	<R0>	0x4001_00EB	<R0>	0x4001_00FB	
0x4001_00CC	TB3MOD	0x4001_00DC	TB3UC	0x4001_00EC	TB3CP1	0x4001_00FC	
0x4001_00CD	<R0>	0x4001_00DD		0x4001_00ED		0x4001_00FD	
0x4001_00CE	<R0>	0x4001_00DE	<R0>	0x4001_00EE	<R0>	0x4001_00FE	
0x4001_00CF	<R0>	0x4001_00DF	<R0>	0x4001_00EF	<R0>	0x4001_00FF	

<TMRB4>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0100	TB4EN	0x4001_0110	TB4FFCR	0x4001_0120	TB4RG0	0x4001_0130	
0x4001_0101	<R0>	0x4001_0111	<R0>	0x4001_0121		0x4001_0131	
0x4001_0102	<R0>	0x4001_0112	<R0>	0x4001_0122	<R0>	0x4001_0132	
0x4001_0103	<R0>	0x4001_0113	<R0>	0x4001_0123	<R0>	0x4001_0133	
0x4001_0104	TB4RUN	0x4001_0114	TB4ST	0x4001_0124	TB4RG1	0x4001_0134	
0x4001_0105	<R0>	0x4001_0115	<R0>	0x4001_0125		0x4001_0135	
0x4001_0106	<R0>	0x4001_0116	<R0>	0x4001_0126	<R0>	0x4001_0136	
0x4001_0107	<R0>	0x4001_0117	<R0>	0x4001_0127	<R0>	0x4001_0137	
0x4001_0108	TB4CR	0x4001_0118	TB4IM	0x4001_0128	TB4CP0	0x4001_0138	
0x4001_0109	<R0>	0x4001_0119	<R0>	0x4001_0129		0x4001_0139	
0x4001_010A	<R0>	0x4001_011A	<R0>	0x4001_012A	<R0>	0x4001_013A	
0x4001_010B	<R0>	0x4001_011B	<R0>	0x4001_012B	<R0>	0x4001_013B	
0x4001_010C	TB4MOD	0x4001_011C	TB4UC	0x4001_012C	TB4CP1	0x4001_013C	
0x4001_010D	<R0>	0x4001_011D		0x4001_012D		0x4001_013D	
0x4001_010E	<R0>	0x4001_011E	<R0>	0x4001_012E	<R0>	0x4001_013E	
0x4001_010F	<R0>	0x4001_011F	<R0>	0x4001_012F	<R0>	0x4001_013F	

<TMRB5>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0140	TB5EN	0x4001_0150	TB5FFCR	0x4001_0160	TB5RG0	0x4001_0170	
0x4001_0141	<R0>	0x4001_0151	<R0>	0x4001_0161		0x4001_0171	
0x4001_0142	<R0>	0x4001_0152	<R0>	0x4001_0162	<R0>	0x4001_0172	
0x4001_0143	<R0>	0x4001_0153	<R0>	0x4001_0163	<R0>	0x4001_0173	
0x4001_0144	TB5RUN	0x4001_0154	TB5ST	0x4001_0164	TB5RG1	0x4001_0174	
0x4001_0145	<R0>	0x4001_0155	<R0>	0x4001_0165		0x4001_0175	
0x4001_0146	<R0>	0x4001_0156	<R0>	0x4001_0166	<R0>	0x4001_0176	
0x4001_0147	<R0>	0x4001_0157	<R0>	0x4001_0167	<R0>	0x4001_0177	
0x4001_0148	TB5CR	0x4001_0158	TB5IM	0x4001_0168	TB5CP0	0x4001_0178	
0x4001_0149	<R0>	0x4001_0159	<R0>	0x4001_0169		0x4001_0179	
0x4001_014A	<R0>	0x4001_015A	<R0>	0x4001_016A	<R0>	0x4001_017A	
0x4001_014B	<R0>	0x4001_015B	<R0>	0x4001_016B	<R0>	0x4001_017B	
0x4001_014C	TB5MOD	0x4001_015C	TB5UC	0x4001_016C	TB5CP1	0x4001_017C	
0x4001_014D	<R0>	0x4001_015D		0x4001_016D		0x4001_017D	
0x4001_014E	<R0>	0x4001_015E	<R0>	0x4001_016E	<R0>	0x4001_017E	
0x4001_014F	<R0>	0x4001_015F	<R0>	0x4001_016F	<R0>	0x4001_017F	

[2] 16-bit timer [3/3]

<TMRB6>

Address	Register name
0x4001_0180	TB6EN
0x4001_0181	<R0>
0x4001_0182	<R0>
0x4001_0183	<R0>
0x4001_0184	TB6RUN
0x4001_0185	<R0>
0x4001_0186	<R0>
0x4001_0187	<R0>
0x4001_0188	TB6CR
0x4001_0189	<R0>
0x4001_018A	<R0>
0x4001_018B	<R0>
0x4001_018C	TB6MOD
0x4001_018D	<R0>
0x4001_018E	<R0>
0x4001_018F	<R0>

Address	Register name
0x4001_0190	TB6FFCR
0x4001_0191	<R0>
0x4001_0192	<R0>
0x4001_0193	<R0>
0x4001_0194	TB6ST
0x4001_0195	<R0>
0x4001_0196	<R0>
0x4001_0197	<R0>
0x4001_0198	TB6IM
0x4001_0199	<R0>
0x4001_019A	<R0>
0x4001_019B	<R0>
0x4001_019C	TB6UC
0x4001_019D	
0x4001_019E	<R0>
0x4001_019F	<R0>

Address	Register name
0x4001_01A0	TB6RG0
0x4001_01A1	
0x4001_01A2	<R0>
0x4001_01A3	<R0>
0x4001_01A4	TB6RG1
0x4001_01A5	
0x4001_01A6	<R0>
0x4001_01A7	<R0>
0x4001_01A8	TB6CP0
0x4001_01A9	
0x4001_01AA	<R0>
0x4001_01AB	<R0>
0x4001_01AC	TB6CP1
0x4001_01AD	
0x4001_01AE	<R0>
0x4001_01AF	<R0>

Address	Register name
0x4001_01B0	
0x4001_01B1	
0x4001_01B2	
0x4001_01B3	
0x4001_01B4	
0x4001_01B5	
0x4001_01B6	
0x4001_01B7	
0x4001_01B8	
0x4001_01B9	
0x4001_01BA	
0x4001_01BB	
0x4001_01BC	
0x4001_01BD	
0x4001_01BE	
0x4001_01BF	

<TMRB7>

Address	Register name
0x4001_01C0	TB7EN
0x4001_01C1	<R0>
0x4001_01C2	<R0>
0x4001_01C3	<R0>
0x4001_01C4	TB7RUN
0x4001_01C5	<R0>
0x4001_01C6	<R0>
0x4001_01C7	<R0>
0x4001_01C8	TB7CR
0x4001_01C9	<R0>
0x4001_01CA	<R0>
0x4001_01CB	<R0>
0x4001_01CC	TB7MOD
0x4001_01CD	<R0>
0x4001_01CE	<R0>
0x4001_01CF	<R0>

Address	Register name
0x4001_01D0	TB7FFCR
0x4001_01D1	<R0>
0x4001_01D2	<R0>
0x4001_01D3	<R0>
0x4001_01D4	TB7ST
0x4001_01D5	<R0>
0x4001_01D6	<R0>
0x4001_01D7	<R0>
0x4001_01D8	TB7IM
0x4001_01D9	<R0>
0x4001_01DA	<R0>
0x4001_01DB	<R0>
0x4001_01DC	TB7UC
0x4001_01DD	
0x4001_01DE	<R0>
0x4001_01DF	<R0>

Address	Register name
0x4001_01E0	TB7RG0
0x4001_01E1	
0x4001_01E2	<R0>
0x4001_01E3	<R0>
0x4001_01E4	TB7RG1
0x4001_01E5	
0x4001_01E6	<R0>
0x4001_01E7	<R0>
0x4001_01E8	TB7CP0
0x4001_01E9	
0x4001_01EA	<R0>
0x4001_01EB	<R0>
0x4001_01EC	TB7CP1
0x4001_01ED	
0x4001_01EE	<R0>
0x4001_01EF	<R0>

Address	Register name
0x4001_01F0	
0x4001_01F1	
0x4001_01F2	
0x4001_01F3	
0x4001_01F4	
0x4001_01F5	
0x4001_01F6	
0x4001_01F7	
0x4001_01F8	
0x4001_01F9	
0x4001_01FA	
0x4001_01FB	
0x4001_01FC	
0x4001_01FD	
0x4001_01FE	
0x4001_01FF	

25.1.3 [3] Encoder input (ENC)

<ENC0>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0400	EN0TNCR	0x4001_0410		0x4001_0420		0x4001_0430	
0x4001_0401		0x4001_0411		0x4001_0421		0x4001_0431	
0x4001_0402		0x4001_0412		0x4001_0422		0x4001_0432	
0x4001_0403	<R0>	0x4001_0413		0x4001_0423		0x4001_0433	
0x4001_0404	EN0RELOAD	0x4001_0414		0x4001_0424		0x4001_0434	
0x4001_0405		0x4001_0415		0x4001_0425		0x4001_0435	
0x4001_0406	<R0>	0x4001_0416		0x4001_0426		0x4001_0436	
0x4001_0407	<R0>	0x4001_0417		0x4001_0427		0x4001_0437	
0x4001_0408	EN0INT	0x4001_0418		0x4001_0428		0x4001_0438	
0x4001_0409		0x4001_0419		0x4001_0429		0x4001_0439	
0x4001_040A		0x4001_041A		0x4001_042A		0x4001_043A	
0x4001_040B	<R0>	0x4001_041B		0x4001_042B		0x4001_043B	
0x4001_040C	EN0CNT	0x4001_041C		0x4001_042C		0x4001_043C	
0x4001_040D		0x4001_041D		0x4001_042D		0x4001_043D	
0x4001_040E		0x4001_041E		0x4001_042E		0x4001_043E	
0x4001_040F	<R0>	0x4001_041F		0x4001_042F		0x4001_043F	

<ENC1>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0500	EN1TNCR	0x4001_0510		0x4001_0520		0x4001_0530	
0x4001_0501		0x4001_0511		0x4001_0521		0x4001_0531	
0x4001_0502		0x4001_0512		0x4001_0522		0x4001_0532	
0x4001_0503	<R0>	0x4001_0513		0x4001_0523		0x4001_0533	
0x4001_0504	EN1RELOAD	0x4001_0514		0x4001_0524		0x4001_0534	
0x4001_0505		0x4001_0515		0x4001_0525		0x4001_0535	
0x4001_0506	<R0>	0x4001_0516		0x4001_0526		0x4001_0536	
0x4001_0507	<R0>	0x4001_0517		0x4001_0527		0x4001_0537	
0x4001_0508	EN1INT	0x4001_0518		0x4001_0528		0x4001_0538	
0x4001_0509		0x4001_0519		0x4001_0529		0x4001_0539	
0x4001_050A		0x4001_051A		0x4001_052A		0x4001_053A	
0x4001_050B	<R0>	0x4001_051B		0x4001_052B		0x4001_053B	
0x4001_050C	EN1CNT	0x4001_051C		0x4001_052C		0x4001_053C	
0x4001_050D		0x4001_051D		0x4001_052D		0x4001_053D	
0x4001_050E		0x4001_051E		0x4001_052E		0x4001_053E	
0x4001_050F	<R0>	0x4001_051F		0x4001_052F		0x4001_053F	

25.1.4 [4] Serial bus interface (SBI)

<SBI0>

Address	Register name
0x4002_0000	SBI0CR0
0x4002_0001	
0x4002_0002	
0x4002_0003	
0x4002_0004	SBI0CR1
0x4002_0005	
0x4002_0006	
0x4002_0007	
0x4002_0008	SBI0DBR
0x4002_0009	
0x4002_000A	
0x4002_000B	
0x4002_000C	SBI0I2CAR
0x4002_000D	
0x4002_000E	
0x4002_000F	

Address	Register name
0x4002_0010	SBI0CR2/SR
0x4002_0011	
0x4002_0012	
0x4002_0013	
0x4002_0014	SBI0BR0
0x4002_0015	
0x4002_0016	
0x4002_0017	
0x4002_0018	
0x4002_0019	
0x4002_001A	
0x4002_001B	
0x4002_001C	
0x4002_001D	
0x4002_001E	
0x4002_001F	

<SBI1>

Address	Register name
0x4002_0020	SBI1CR0
0x4002_0021	
0x4002_0022	
0x4002_0023	
0x4002_0024	SBI1CR1
0x4002_0025	
0x4002_0026	
0x4002_0027	
0x4002_0028	SBI1DBR
0x4002_0029	
0x4002_002A	
0x4002_002B	
0x4002_002C	SBI1I2CAR
0x4002_002D	
0x4002_002E	
0x4002_002F	

Address	Register name
0x4002_0030	SBI1CR2/SR
0x4002_0031	
0x4002_0032	
0x4002_0033	
0x4002_0034	SBI1BR0
0x4002_0035	
0x4002_0036	
0x4002_0037	
0x4002_0038	
0x4002_0039	
0x4002_003A	
0x4002_003B	
0x4002_003C	
0x4002_003D	"
0x4002_003E	
0x4002_003F	

Address	Register name
0x4002_0040	
0x4002_0041	
0x4002_0042	
0x4002_0043	
0x4002_0044	
0x4002_0045	
0x4002_0046	
0x4002_0047	
0x4002_0048	
0x4002_0049	
0x4002_004A	
0x4002_004B	
0x4002_004C	
0x4002_004D	
0x4002_004E	
0x4002_004F	

Address	Register name
0x4002_0050	
0x4002_0051	
0x4002_0052	
0x4002_0053	
0x4002_0054	
0x4002_0055	
0x4002_0056	
0x4002_0057	
0x4002_0058	
0x4002_0059	
0x4002_005A	
0x4002_005B	
0x4002_005C	
0x4002_005D	
0x4002_005E	
0x4002_005F	

Address	Register name
0x4002_0060	
0x4002_0061	
0x4002_0062	
0x4002_0063	
0x4002_0064	
0x4002_0065	
0x4002_0066	
0x4002_0067	
0x4002_0068	
0x4002_0069	
0x4002_006A	
0x4002_006B	
0x4002_006C	
0x4002_006D	
0x4002_006E	
0x4002_006F	

Address	Register name
0x4002_0070	
0x4002_0071	
0x4002_0072	
0x4002_0073	
0x4002_0074	
0x4002_0075	
0x4002_0076	
0x4002_0077	
0x4002_0078	
0x4002_0079	
0x4002_007A	
0x4002_007B	
0x4002_007C	
0x4002_007D	
0x4002_007E	
0x4002_007F	

25.1.5 [5] Serial interface (UART/SIO) [1/2]

<SIO0>

Address	Register name
0x4002_0080	SC0EN
0x4002_0081	<R0>
0x4002_0082	<R0>
0x4002_0083	<R0>
0x4002_0084	SC0BUF
0x4002_0085	<R0>
0x4002_0086	<R0>
0x4002_0087	<R0>
0x4002_0088	SC0CR
0x4002_0089	<R0>
0x4002_008A	<R0>
0x4002_008B	<R0>
0x4002_008C	SC0MOD0
0x4002_008D	<R0>
0x4002_008E	<R0>
0x4002_008F	<R0>

Address	Register name
0x4002_0090	SC0BRCR
0x4002_0091	<R0>
0x4002_0092	<R0>
0x4002_0093	<R0>
0x4002_0094	SC0BRADD
0x4002_0095	<R0>
0x4002_0096	<R0>
0x4002_0097	<R0>
0x4002_0098	SC0MOD1
0x4002_0099	<R0>
0x4002_009A	<R0>
0x4002_009B	<R0>
0x4002_009C	SC0MOD2
0x4002_009D	<R0>
0x4002_009E	<R0>
0x4002_009F	<R0>

Address	Register name
0x4002_00A0	SC0RFC
0x4002_00A1	<R0>
0x4002_00A2	<R0>
0x4002_00A3	<R0>
0x4002_00A4	SC0TFC
0x4002_00A5	<R0>
0x4002_00A6	<R0>
0x4002_00A7	<R0>
0x4002_00A8	SC0RST
0x4002_00A9	<R0>
0x4002_00AA	<R0>
0x4002_00AB	<R0>
0x4002_00AC	SC0TST
0x4002_00AD	<R0>
0x4002_00AE	<R0>
0x4002_00AF	<R0>

Address	Register name
0x4002_00B0	SC0FCNF
0x4002_00B1	<R0>
0x4002_00B2	<R0>
0x4002_00B3	<R0>
0x4002_00B4	
0x4002_00B5	
0x4002_00B6	
0x4002_00B7	
0x4002_00B8	
0x4002_00B9	
0x4002_00BA	
0x4002_00BB	
0x4002_00BC	
0x4002_00BD	
0x4002_00BE	
0x4002_00BF	

<SIO1>

Address	Register name
0x4002_00C0	SC1EN
0x4002_00C1	<R0>
0x4002_00C2	<R0>
0x4002_00C3	<R0>
0x4002_00C4	SC1BUF
0x4002_00C5	<R0>
0x4002_00C6	<R0>
0x4002_00C7	<R0>
0x4002_00C8	SC1CR
0x4002_00C9	<R0>
0x4002_00CA	<R0>
0x4002_00CB	<R0>
0x4002_00CC	SC1MOD0
0x4002_00CD	<R0>
0x4002_00CE	<R0>
0x4002_00CF	<R0>

Address	Register name
0x4002_00D0	SC1BRCR
0x4002_00D1	<R0>
0x4002_00D2	<R0>
0x4002_00D3	<R0>
0x4002_00D4	SC1BRADD
0x4002_00D5	<R0>
0x4002_00D6	<R0>
0x4002_00D7	<R0>
0x4002_00D8	SC1MOD1
0x4002_00D9	<R0>
0x4002_00DA	<R0>
0x4002_00DB	<R0>
0x4002_00DC	SC1MOD2
0x4002_00DD	<R0>
0x4002_00DE	<R0>
0x4002_00DF	<R0>

Address	Register name
0x4002_00E0	SC1RFC
0x4002_00E1	<R0>
0x4002_00E2	<R0>
0x4002_00E3	<R0>
0x4002_00E4	SC1TFC
0x4002_00E5	<R0>
0x4002_00E6	<R0>
0x4002_00E7	<R0>
0x4002_00E8	SC1RST
0x4002_00E9	<R0>
0x4002_00EA	<R0>
0x4002_00EB	<R0>
0x4002_00EC	SC1TST
0x4002_00ED	<R0>
0x4002_00EE	<R0>
0x4002_00EF	<R0>

Address	Register name
0x4002_00F0	SC1FCNF
0x4002_00F1	<R0>
0x4002_00F2	<R0>
0x4002_00F3	<R0>
0x4002_00F4	
0x4002_00F5	
0x4002_00F6	
0x4002_00F7	
0x4002_00F8	
0x4002_00F9	
0x4002_00FA	
0x4002_00FB	
0x4002_00FC	
0x4002_00FD	
0x4002_00FE	
0x4002_00FF	

<SIO2>

Address	Register name
0x4002_0100	SC2EN
0x4002_0101	<R0>
0x4002_0102	<R0>
0x4002_0103	<R0>
0x4002_0104	SC2BUF
0x4002_0105	<R0>
0x4002_0106	<R0>
0x4002_0107	<R0>
0x4002_0108	SC2CR
0x4002_0109	<R0>
0x4002_010A	<R0>
0x4002_010B	<R0>
0x4002_010C	SC2MOD0
0x4002_010D	<R0>
0x4002_010E	<R0>
0x4002_010F	<R0>

Address	Register name
0x4002_0110	SC2BRCR
0x4002_0111	<R0>
0x4002_0112	<R0>
0x4002_0113	<R0>
0x4002_0114	SC2BRADD
0x4002_0115	<R0>
0x4002_0116	<R0>
0x4002_0117	<R0>
0x4002_0118	SC2MOD1
0x4002_0119	<R0>
0x4002_011A	<R0>
0x4002_011B	<R0>
0x4002_011C	SC2MOD2
0x4002_011D	<R0>
0x4002_011E	<R0>
0x4002_011F	<R0>

Address	Register name
0x4002_0120	SC2RFC
0x4002_0121	<R0>
0x4002_0122	<R0>
0x4002_0123	<R0>
0x4002_0124	SC2TFC
0x4002_0125	<R0>
0x4002_0126	<R0>
0x4002_0127	<R0>
0x4002_0128	SC2RST
0x4002_0129	<R0>
0x4002_012A	<R0>
0x4002_012B	<R0>
0x4002_012C	SC2TST
0x4002_012D	<R0>
0x4002_012E	<R0>
0x4002_012F	<R0>

Address	Register name
0x4002_0130	SC2FCNF
0x4002_0131	<R0>
0x4002_0132	<R0>
0x4002_0133	<R0>
0x4002_0134	
0x4002_0135	
0x4002_0136	
0x4002_0137	
0x4002_0138	
0x4002_0139	
0x4002_013A	
0x4002_013B	
0x4002_013C	
0x4002_013D	
0x4002_013E	
0x4002_013F	

[5] Serial interface (UART/SIO) [2/2]

<SIO3>

Address	Register name
0x4002_0140	SC3EN
0x4002_0141	<R0>
0x4002_0142	<R0>
0x4002_0143	<R0>
0x4002_0144	SC3BUF
0x4002_0145	<R0>
0x4002_0146	<R0>
0x4002_0147	<R0>
0x4002_0148	SC3CR
0x4002_0149	<R0>
0x4002_014A	<R0>
0x4002_014B	<R0>
0x4002_014C	SC3MOD0
0x4002_014D	<R0>
0x4002_014E	<R0>
0x4002_014F	<R0>

Address	Register name
0x4002_0150	SC3BRCR
0x4002_0151	<R0>
0x4002_0152	<R0>
0x4002_0153	<R0>
0x4002_0154	SC3BRADD
0x4002_0155	<R0>
0x4002_0156	<R0>
0x4002_0157	<R0>
0x4002_0158	SC3MOD1
0x4002_0159	<R0>
0x4002_015A	<R0>
0x4002_015B	<R0>
0x4002_015C	SC3MOD2
0x4002_015D	<R0>
0x4002_015E	<R0>
0x4002_015F	<R0>

Address	Register name
0x4002_0160	SC3RFC
0x4002_0161	<R0>
0x4002_0162	<R0>
0x4002_0163	<R0>
0x4002_0164	SC3TFC
0x4002_0165	<R0>
0x4002_0166	<R0>
0x4002_0167	<R0>
0x4002_0168	SC3RST
0x4002_0169	<R0>
0x4002_016A	<R0>
0x4002_016B	<R0>
0x4002_016C	SC3TST
0x4002_016D	<R0>
0x4002_016E	<R0>
0x4002_016F	<R0>

Address	Register name
0x4002_0170	SC3FCNF
0x4002_0171	<R0>
0x4002_0172	<R0>
0x4002_0173	<R0>
0x4002_0174	
0x4002_0175	
0x4002_0176	
0x4002_0177	
0x4002_0178	
0x4002_0179	
0x4002_017A	
0x4002_017B	
0x4002_017C	
0x4002_017D	
0x4002_017E	
0x4002_017F	

<SIO4>

Address	Register name
0x4002_0180	SC4EN
0x4002_0181	<R0>
0x4002_0182	<R0>
0x4002_0183	<R0>
0x4002_0184	SC4BUF
0x4002_0185	<R0>
0x4002_0186	<R0>
0x4002_0187	<R0>
0x4002_0188	SC4CR
0x4002_0189	<R0>
0x4002_018A	<R0>
0x4002_018B	<R0>
0x4002_018C	SC4MOD0
0x4002_018D	<R0>
0x4002_018E	<R0>
0x4002_018F	<R0>

Address	Register name
0x4002_0190	SC4BRCR
0x4002_0191	<R0>
0x4002_0192	<R0>
0x4002_0193	<R0>
0x4002_0194	SC4BRADD
0x4002_0195	<R0>
0x4002_0196	<R0>
0x4002_0197	<R0>
0x4002_0198	SC4MOD1
0x4002_0199	<R0>
0x4002_019A	<R0>
0x4002_019B	<R0>
0x4002_019C	SC4MOD2
0x4002_019D	<R0>
0x4002_019E	<R0>
0x4002_019F	<R0>

Address	Register name
0x4002_01A0	SC4RFC
0x4002_01A1	<R0>
0x4002_01A2	<R0>
0x4002_01A3	<R0>
0x4002_01A4	SC4TFC
0x4002_01A5	<R0>
0x4002_01A6	<R0>
0x4002_01A7	<R0>
0x4002_01A8	SC4RST
0x4002_01A9	<R0>
0x4002_01AA	<R0>
0x4002_01AB	<R0>
0x4002_01AC	SC4TST
0x4002_01AD	<R0>
0x4002_01AE	<R0>
0x4002_01AF	<R0>

Address	Register name
0x4002_01B0	SC4FCNF
0x4002_01B1	<R0>
0x4002_01B2	<R0>
0x4002_01B3	<R0>
0x4002_01B4	
0x4002_01B5	
0x4002_01B6	
0x4002_01B7	
0x4002_01B8	
0x4002_01B9	
0x4002_01BA	
0x4002_01BB	
0x4002_01BC	
0x4002_01BD	
0x4002_01BE	
0x4002_01BF	

25.1.6 [6] 12-bit A/D converter (A/DC) [1/2]

<ADC>

Address	Register name
0x4003_0000	ADCLK
0x4003_0001	
0x4003_0002	
0x4003_0003	
0x4003_0004	ADMOD0
0x4003_0005	
0x4003_0006	
0x4003_0007	
0x4003_0008	ADMOD1
0x4003_0009	
0x4003_000A	
0x4003_000B	
0x4003_000C	ADMOD2
0x4003_000D	
0x4003_000E	
0x4003_000F	

Address	Register name
0x4003_0010	ADCMPCR0
0x4003_0011	
0x4003_0012	
0x4003_0013	
0x4003_0014	ADCMPCR1
0x4003_0015	
0x4003_0016	
0x4003_0017	
0x4003_0018	ADCMP0
0x4003_0019	
0x4003_001A	
0x4003_001B	
0x4003_001C	ADCMP1
0x4003_001D	
0x4003_001E	
0x4003_001F	

Address	Register name
0x4003_0020	ADREG0
0x4003_0021	
0x4003_0022	
0x4003_0023	
0x4003_0024	ADREG1
0x4003_0025	
0x4003_0026	
0x4003_0027	
0x4003_0028	ADREG2
0x4003_0029	
0x4003_002A	
0x4003_002B	
0x4003_002C	ADREG3
0x4003_002D	
0x4003_002E	
0x4003_002F	

Address	Register name
0x4003_0030	ADREG4
0x4003_0031	
0x4003_0032	
0x4003_0033	
0x4003_0034	ADREG5
0x4003_0035	
0x4003_0036	
0x4003_0037	
0x4003_0038	ADREG6
0x4003_0039	
0x4003_003A	
0x4003_003B	
0x4003_003C	ADREG7
0x4003_003D	
0x4003_003E	
0x4003_003F	

Address	Register name
0x4003_0040	ADREG8
0x4003_0041	
0x4003_0042	
0x4003_0043	
0x4003_0044	ADREG9
0x4003_0045	
0x4003_0046	
0x4003_0047	
0x4003_0048	ADREG10
0x4003_0049	
0x4003_004A	
0x4003_004B	
0x4003_004C	ADREG11
0x4003_004D	
0x4003_004E	
0x4003_004F	

Address	Register name
0x4003_0050	ADPSEL0
0x4003_0051	
0x4003_0052	
0x4003_0053	
0x4003_0054	ADPSEL1
0x4003_0055	
0x4003_0056	
0x4003_0057	
0x4003_0058	ADPSEL2
0x4003_0059	
0x4003_005A	
0x4003_005B	
0x4003_005C	ADPSEL3
0x4003_005D	
0x4003_005E	
0x4003_005F	

Address	Register name
0x4003_0060	Reserved
0x4003_0061	
0x4003_0062	
0x4003_0063	
0x4003_0064	Reserved
0x4003_0065	
0x4003_0066	
0x4003_0067	
0x4003_0068	Reserved
0x4003_0069	
0x4003_006A	
0x4003_006B	
0x4003_006C	Reserved
0x4003_006D	
0x4003_006E	
0x4003_006F	

Address	Register name
0x4003_0070	Reserved
0x4003_0071	
0x4003_0072	
0x4003_0073	
0x4003_0074	Reserved
0x4003_0075	
0x4003_0076	
0x4003_0077	
0x4003_0078	Reserved
0x4003_0079	
0x4003_007A	
0x4003_007B	
0x4003_007C	Reserved
0x4003_007D	
0x4003_007E	
0x4003_007F	

Address	Register name
0x4003_0080	ADPINTS0
0x4003_0081	
0x4003_0082	
0x4003_0083	
0x4003_0084	ADPINTS1
0x4003_0085	
0x4003_0086	
0x4003_0087	
0x4003_0088	ADPINTS2
0x4003_0089	
0x4003_008A	
0x4003_008B	
0x4003_008C	ADPINTS3
0x4003_008D	
0x4003_008E	
0x4003_008F	

Address	Register name
0x4003_0090	ADPINTS4
0x4003_0091	
0x4003_0092	
0x4003_0093	
0x4003_0094	ADPINTS5
0x4003_0095	
0x4003_0096	
0x4003_0097	
0x4003_0098	ADPSET0
0x4003_0099	
0x4003_009A	
0x4003_009B	
0x4003_009C	ADPSET1
0x4003_009D	
0x4003_009E	
0x4003_009F	

Address	Register name
0x4003_00A0	ADPSET2
0x4003_00A1	
0x4003_00A2	
0x4003_00A3	
0x4003_00A4	ADPSET3
0x4003_00A5	
0x4003_00A6	
0x4003_00A7	
0x4003_00A8	ADPSET4
0x4003_00A9	
0x4003_00AA	
0x4003_00AB	
0x4003_00AC	ADPSET5
0x4003_00AD	
0x4003_00AE	
0x4003_00AF	

Address	Register name
0x4003_00B0	ADTSET03
0x4003_00B1	
0x4003_00B2	
0x4003_00B3	
0x4003_00B4	ADTSET47
0x4003_00B5	
0x4003_00B6	
0x4003_00B7	
0x4003_00B8	ADTSET811
0x4003_00B9	
0x4003_00BA	
0x4003_00BB	
0x4003_00BC	ADSSET03
0x4003_00BD	
0x4003_00BE	
0x4003_00BF	

[6] 12-bit A/D converter (A/DC) [2/2]

Address	Register name
0x4003_00C0	ADSSET47
0x4003_00C1	
0x4003_00C2	
0x4003_00C3	
0x4003_00C4	ADSSET811
0x4003_00C5	
0x4003_00C6	
0x4003_00C7	
0x4003_00C8	ADASET03
0x4003_00C9	
0x4003_00CA	
0x4003_00CB	
0x4003_00CC	ADASET47
0x4003_00CD	
0x4003_00CE	
0x4003_00CF	

Address	Register name
0x4003_00D0	ADASET811
0x4003_00D1	
0x4003_00D2	
0x4003_00D3	
0x4003_00D4	ADMOD3
0x4003_00D5	
0x4003_00D6	
0x4003_00D7	
0x4003_00D8	
0x4003_00D9	
0x4003_00DA	
0x4003_00DB	
0x4003_00DC	
0x4003_00DD	
0x4003_00DE	
0x4003_00DF	

Address	Register name
0x4003_00E0	
0x4003_00E1	
0x4003_00E2	
0x4003_00E3	
0x4003_00E4	
0x4003_00E5	
0x4003_00E6	
0x4003_00E7	
0x4003_00E8	
0x4003_00E9	
0x4003_00EA	
0x4003_00EB	
0x4003_00EC	
0x4003_00ED	
0x4003_00EE	
0x4003_00EF	

Address	Register name
0x4003_00F0	
0x4003_00F1	
0x4003_00F2	
0x4003_00F3	
0x4003_00F4	
0x4003_00F5	
0x4003_00F6	
0x4003_00F7	
0x4003_00F8	
0x4003_00F9	
0x4003_00FA	
0x4003_00FB	
0x4003_00FC	
0x4003_00FD	
0x4003_00FE	
0x4003_00FF	

25.1.7 [7] Watchdog timer (WDT)

Address	Register name
0x4004_0000	WDMOD
0x4004_0001	<R0>
0x4004_0002	<R0>
0x4004_0003	<R0>
0x4004_0004	WDCR
0x4004_0005	<R0>
0x4004_0006	<R0>
0x4004_0007	<R0>
0x4004_0008	
0x4004_0009	
0x4004_000A	
0x4004_000B	
0x4004_000C	
0x4004_000D	
0x4004_000E	
0x4004_000F	

Address	Register name
0x4004_0010	
0x4004_0011	
0x4004_0012	
0x4004_0013	
0x4004_0014	
0x4004_0015	
0x4004_0016	
0x4004_0017	
0x4004_0018	
0x4004_0019	
0x4004_001A	
0x4004_001B	
0x4004_001C	
0x4004_001D	
0x4004_001E	
0x4004_001F	

Address	Register name
0x4004_0020	
0x4004_0021	
0x4004_0022	
0x4004_0023	
0x4004_0024	
0x4004_0025	
0x4004_0026	
0x4004_0027	
0x4004_0028	
0x4004_0029	
0x4004_002A	
0x4004_002B	
0x4004_002C	
0x4004_002D	
0x4004_002E	
0x4004_002F	

Address	Register name
0x4004_0030	
0x4004_0031	
0x4004_0032	
0x4004_0033	
0x4004_0034	
0x4004_0035	
0x4004_0036	
0x4004_0037	
0x4004_0038	
0x4004_0039	
0x4004_003A	
0x4004_003B	
0x4004_003C	
0x4004_003D	
0x4004_003E	
0x4004_003F	

25.1.8 [8] Real time clock (RTC)

Address	Register name
0x4004_0100	SECR
0x4004_0101	MINR
0x4004_0102	HOURR
0x4004_0103	
0x4004_0104	DAYR
0x4004_0105	DATER
0x4004_0106	MONTHR
0x4004_0107	YEARR
0x4004_0108	PAGER
0x4004_0109	
0x4004_010A	
0x4004_010B	
0x4004_010C	RESTR
0x4004_010D	
0x4004_010E	
0x4004_010F	

Address	Register name
0x4004_0110	
0x4004_0111	
0x4004_0112	
0x4004_0113	
0x4004_0114	
0x4004_0115	
0x4004_0116	
0x4004_0117	
0x4004_0118	
0x4004_0119	
0x4004_011A	
0x4004_011B	
0x4004_011C	
0x4004_011D	
0x4004_011E	
0x4004_011F	

Address	Register name
0x4004_0120	
0x4004_0121	
0x4004_0122	
0x4004_0123	
0x4004_0124	
0x4004_0125	
0x4004_0126	
0x4004_0127	
0x4004_0128	
0x4004_0129	
0x4004_012A	
0x4004_012B	
0x4004_012C	
0x4004_012D	
0x4004_012E	
0x4004_012F	

Address	Register name
0x4004_0130	
0x4004_0131	
0x4004_0132	
0x4004_0133	
0x4004_0134	
0x4004_0135	
0x4004_0136	
0x4004_0137	
0x4004_0138	
0x4004_0139	
0x4004_013A	
0x4004_013B	
0x4004_013C	
0x4004_013D	
0x4004_013E	
0x4004_013F	

25.1.9 [9] Clock generator (CG)

Address	Register name
0x4004_0200	CGSYSCR
0x4004_0201	
0x4004_0202	
0x4004_0203	<R0>
0x4004_0204	CGOSCCR
0x4004_0205	
0x4004_0206	
0x4004_0207	
0x4004_0208	CGSTBYCR
0x4004_0209	
0x4004_020A	
0x4004_020B	<R0>
0x4004_020C	CGPLLSEL
0x4004_020D	
0x4004_020E	<R0>
0x4004_020F	<R0>

Address	Register name
0x4004_0210	CGCKSEL
0x4004_0211	<R0>
0x4004_0212	<R0>
0x4004_0213	<R0>
0x4004_0214	CGICRCG
0x4004_0215	<R0>
0x4004_0216	<R0>
0x4004_0217	<R0>
0x4004_0218	CGNMIFLG
0x4004_0219	<R0>
0x4004_021A	<R0>
0x4004_021B	<R0>
0x4004_021C	CGRSTFLG
0x4004_021D	<R0>
0x4004_021E	<R0>
0x4004_021F	<R0>

Address	Register name
0x4004_0220	CGIMCGA
0x4004_0221	
0x4004_0222	
0x4004_0223	
0x4004_0224	CGIMCGB
0x4004_0225	
0x4004_0226	
0x4004_0227	
0x4004_0228	CGIMCGC
0x4004_0229	
0x4004_022A	
0x4004_022B	
0x4004_022C	CGIMCGD
0x4004_022D	
0x4004_022E	
0x4004_022F	

Address	Register name
0x4004_0230	CGIMCGE
0x4004_0231	
0x4004_0232	
0x4004_0233	
0x4004_0234	
0x4004_0235	
0x4004_0236	
0x4004_0237	
0x4004_0238	
0x4004_0239	
0x4004_023A	
0x4004_023B	
0x4004_023C	
0x4004_023D	
0x4004_023E	
0x4004_023F	

25.1.10 [10] Remote control signal preprocessor (RMC)

Address	Register name
0x4004_0400	RMCEN
0x4004_0401	<R0>
0x4004_0402	<R0>
0x4004_0403	<R0>
0x4004_0404	RMCREN
0x4004_0405	<R0>
0x4004_0406	<R0>
0x4004_0407	<R0>
0x4004_0408	RMCRBUF1
0x4004_0409	
0x4004_040A	
0x4004_040B	
0x4004_040C	RMCRBUF2
0x4004_040D	
0x4004_040E	
0x4004_040F	

Address	Register name
0x4004_0410	RMCRBUF3
0x4004_0411	
0x4004_0412	
0x4004_0413	
0x4004_0414	RMCRCR1
0x4004_0415	
0x4004_0416	
0x4004_0417	
0x4004_0418	RMCRCR2
0x4004_0419	
0x4004_041A	
0x4004_041B	
0x4004_041C	RMCRCR3
0x4004_041D	
0x4004_041E	<R0>
0x4004_041F	<R0>

Address	Register name
0x4004_0420	RMCCR4
0x4004_0421	<R0>
0x4004_0422	<R0>
0x4004_0423	<R0>
0x4004_0424	RMCRSTAT
0x4004_0425	
0x4004_0426	<R0>
0x4004_0427	<R0>
0x4004_0428	RMCEND1
0x4004_0429	
0x4004_042A	
0x4004_042B	
0x4004_042C	RMCEND2
0x4004_042D	
0x4004_042E	
0x4004_042F	

Address	Register name
0x4004_0430	RMCEND3
0x4004_0431	
0x4004_0432	
0x4004_0433	
0x4004_0434	RMCFSSEL
0x4004_0435	
0x4004_0436	
0x4004_0437	
0x4004_0438	
0x4004_0439	
0x4004_043A	
0x4004_043B	
0x4004_043C	
0x4004_043D	
0x4004_043E	
0x4004_043F	

25.1.11 [11] Oscillation frequency detector (OFD)

Address	Register name
0x4004_0800	OFDCR1
0x4004_0801	
0x4004_0802	
0x4004_0803	
0x4004_0804	OFDCR2
0x4004_0805	
0x4004_0806	
0x4004_0807	
0x4004_0808	OFDMN
0x4004_0809	
0x4004_080A	
0x4004_080B	
0x4004_080C	
0x4004_080D	
0x4004_080E	
0x4004_080F	

Address	Register name
0x4004_0810	OFDMX
0x4004_0811	
0x4004_0812	
0x4004_0813	
0x4004_0814	
0x4004_0815	
0x4004_0816	
0x4004_0817	
0x4004_0818	OFDRST
0x4004_0819	
0x4004_081A	
0x4004_081B	
0x4004_081C	OFDSTAT
0x4004_081D	
0x4004_081E	
0x4004_081F	

Address	Register name
0x4004_0820	
0x4004_0821	
0x4004_0822	
0x4004_0823	
0x4004_0824	
0x4004_0825	
0x4004_0826	
0x4004_0827	
0x4004_0828	
0x4004_0829	
0x4004_082A	
0x4004_082B	
0x4004_082C	
0x4004_082D	
0x4004_082E	
0x4004_082F	

Address	Register name
0x4004_0830	
0x4004_0831	
0x4004_0832	
0x4004_0833	
0x4004_0834	
0x4004_0835	
0x4004_0836	
0x4004_0837	
0x4004_0838	
0x4004_0839	
0x4004_083A	
0x4004_083B	
0x4004_083C	
0x4004_083D	
0x4004_083E	
0x4004_083F	

25.1.12 [12] Power on reset (POR), Voltage detecting circuit (VLTD)

Address	Register name
0x4004_0900	VDCR
0x4004_0901	
0x4004_0902	
0x4004_0903	
0x4004_0904	VDSR
0x4004_0905	
0x4004_0906	
0x4004_0907	
0x4004_0908	
0x4004_0909	
0x4004_090A	
0x4004_090B	
0x4004_090C	
0x4004_090D	
0x4004_090E	
0x4004_090F	

Address	Register name
0x4004_0910	
0x4004_0911	
0x4004_0912	
0x4004_0913	
0x4004_0914	
0x4004_0915	
0x4004_0916	
0x4004_0917	
0x4004_0918	
0x4004_0919	
0x4004_091A	
0x4004_091B	
0x4004_091C	
0x4004_091D	
0x4004_091E	
0x4004_091F	

Address	Register name
0x4004_0920	
0x4004_0921	
0x4004_0922	
0x4004_0923	
0x4004_0924	
0x4004_0925	
0x4004_0926	
0x4004_0927	
0x4004_0928	
0x4004_0929	
0x4004_092A	
0x4004_092B	
0x4004_092C	
0x4004_092D	
0x4004_092E	
0x4004_092F	

Address	Register name
0x4004_0930	
0x4004_0931	
0x4004_0932	
0x4004_0933	
0x4004_0934	
0x4004_0935	
0x4004_0936	
0x4004_0937	
0x4004_0938	
0x4004_0939	
0x4004_093A	
0x4004_093B	
0x4004_093C	
0x4004_093D	
0x4004_093E	
0x4004_093F	

25.1.13 [13] Multi purpose timer (PMD)[1/2]

Address	Register name
0x4005_0400	MTPD0MDEN
0x4005_0401	<R0>
0x4005_0402	<R0>
0x4005_0403	<R0>
0x4005_0404	MTPD0PORTMD
0x4005_0405	<R0>
0x4005_0406	<R0>
0x4005_0407	<R0>
0x4005_0408	MTPD0MDCR
0x4005_0409	<R0>
0x4005_040A	<R0>
0x4005_040B	<R0>
0x4005_040C	MTPD0CNTSTA
0x4005_040D	<R0>
0x4005_040E	<R0>
0x4005_040F	<R0>

Address	Register name
0x4005_0410	MTPD0MDCNT
0x4005_0411	
0x4005_0412	<R0>
0x4005_0413	<R0>
0x4005_0414	MTPD0MDPRD
0x4005_0415	
0x4005_0416	<R0>
0x4005_0417	<R0>
0x4005_0418	MTPD0CMPU
0x4005_0419	
0x4005_041A	<R0>
0x4005_041B	<R0>
0x4005_041C	MTPD0CMPV
0x4005_041D	
0x4005_041E	<R0>
0x4005_041F	<R0>

Address	Register name
0x4005_0420	MTPD0CMPW
0x4005_0421	
0x4005_0422	<R0>
0x4005_0423	<R0>
0x4005_0424	Reserved
0x4005_0425	Reserved
0x4005_0426	Reserved
0x4005_0427	Reserved
0x4005_0428	MTPD0MDOUT
0x4005_0429	
0x4005_042A	<R0>
0x4005_042B	<R0>
0x4005_042C	MTPD0MDPOT
0x4005_042D	<R0>
0x4005_042E	<R0>
0x4005_042F	<R0>

Address	Register name
0x4005_0430	MTPD0EMGREL
0x4005_0431	<R0>
0x4005_0432	<R0>
0x4005_0433	<R0>
0x4005_0434	MTPD0EMGCR
0x4005_0435	
0x4005_0436	<R0>
0x4005_0437	<R0>
0x4005_0438	MTPD0EMGST
0x4005_0439	<R0>
0x4005_043A	<R0>
0x4005_043B	<R0>
0x4005_043C	Reserved
0x4005_043D	Reserved
0x4005_043E	Reserved
0x4005_043F	Reserved

Address	Register name
0x4005_0440	Reserved
0x4005_0441	Reserved
0x4005_0442	Reserved
0x4005_0443	Reserved
0x4005_0444	MTPD0DTR
0x4005_0445	<R0>
0x4005_0446	<R0>
0x4005_0447	<R0>
0x4005_0448	MTPD0TRGCMP0
0x4005_0449	
0x4005_044A	<R0>
0x4005_044B	<R0>
0x4005_044C	MTPD0TRGCMP1
0x4005_044D	
0x4005_044E	<R0>
0x4005_044F	<R0>

Address	Register name
0x4005_0450	Reserved
0x4005_0451	Reserved
0x4005_0452	Reserved
0x4005_0453	Reserved
0x4005_0454	Reserved
0x4005_0455	Reserved
0x4005_0456	Reserved
0x4005_0457	Reserved
0x4005_0458	MTPD0TRGCR
0x4005_0459	
0x4005_045A	<R0>
0x4005_045B	<R0>
0x4005_045C	MTPD0TRGMD
0x4005_045D	<R0>
0x4005_045E	<R0>
0x4005_045F	<R0>

Address	Register name
0x4005_0460	Reserved
0x4005_0461	Reserved
0x4005_0462	Reserved
0x4005_0463	Reserved
0x4005_0464	Reserved
0x4005_0465	
0x4005_0466	
0x4005_0467	
0x4005_0468	Reserved
0x4005_0469	
0x4005_046A	
0x4005_046B	
0x4005_046C	Reserved
0x4005_046D	
0x4005_046E	
0x4005_046F	

Address	Register name
0x4005_0470	Reserved
0x4005_0471	
0x4005_0472	
0x4005_0473	
0x4005_0474	Reserved
0x4005_0475	
0x4005_0476	
0x4005_0477	
0x4005_0478	Reserved
0x4005_0479	
0x4005_047A	
0x4005_047B	
0x4005_047C	Reserved
0x4005_047D	
0x4005_047E	
0x4005_047F	

[13] Multi purpose timer (PMD) [2/2]

Address	Register name
0x4005_0480	MTPD1MDEN
0x4005_0481	<R0>
0x4005_0482	<R0>
0x4005_0483	<R0>
0x4005_0484	MTPD1PORTMD
0x4005_0485	<R0>
0x4005_0486	<R0>
0x4005_0487	<R0>
0x4005_0488	MTPD1MDCR
0x4005_0489	<R0>
0x4005_048A	<R0>
0x4005_048B	<R0>
0x4005_048C	MTPD1CNTSTA
0x4005_048D	<R0>
0x4005_048E	<R0>
0x4005_048F	<R0>

Address	Register name
0x4005_0490	MTPD1MDCNT
0x4005_0491	
0x4005_0492	<R0>
0x4005_0493	<R0>
0x4005_0494	MTPD1MDPRD
0x4005_0495	
0x4005_0496	<R0>
0x4005_0497	<R0>
0x4005_0498	MTPD1CMPU
0x4005_0499	
0x4005_049A	<R0>
0x4005_049B	<R0>
0x4005_049C	MTPD1CMPV
0x4005_049D	
0x4005_049E	<R0>
0x4005_049F	<R0>

Address	Register name
0x4005_04A0	MTPD1CMPW
0x4005_04A1	
0x4005_04A2	<R0>
0x4005_04A3	<R0>
0x4005_04A4	Reserved
0x4005_04A5	Reserved
0x4005_04A6	Reserved
0x4005_04A7	Reserved
0x4005_04A8	MTPD1MDOUT
0x4005_04A9	
0x4005_04AA	<R0>
0x4005_04AB	<R0>
0x4005_04AC	MTPD1MDPOT
0x4005_04AD	<R0>
0x4005_04AE	<R0>
0x4005_04AF	<R0>

Address	Register name
0x4005_04B0	MTPD1EMGREL
0x4005_04B1	<R0>
0x4005_04B2	<R0>
0x4005_04B3	<R0>
0x4005_04B4	MTPD1EMGCR
0x4005_04B5	
0x4005_04B6	<R0>
0x4005_04B7	<R0>
0x4005_04B8	MTPD1EMGST
0x4005_04B9	<R0>
0x4005_04BA	<R0>
0x4005_04BB	<R0>
0x4005_04BC	Reserved
0x4005_04BD	
0x4005_04BE	
0x4005_04BF	

Address	Register name
0x4005_04C0	Reserved
0x4005_04C1	
0x4005_04C2	
0x4005_04C3	
0x4005_04C4	MTPD1TR
0x4005_04C5	<R0>
0x4005_04C6	<R0>
0x4005_04C7	<R0>
0x4005_04C8	MTPD1TRGCMP0
0x4005_04C9	
0x4005_04CA	<R0>
0x4005_04CB	<R0>
0x4005_04CC	MTPD1TRGCMP1
0x4005_04CD	
0x4005_04CE	<R0>
0x4005_04CF	<R0>

Address	Register name
0x4005_04D0	Reserved
0x4005_04D1	
0x4005_04D2	
0x4005_04D3	
0x4005_04D4	Reserved
0x4005_04D5	
0x4005_04D6	
0x4005_04D7	
0x4005_04D8	MTPD1TRGCR
0x4005_04D9	
0x4005_04DA	<R0>
0x4005_04DB	<R0>
0x4005_04DC	MTPD1TRGMD
0x4005_04DD	<R0>
0x4005_04DE	<R0>
0x4005_04DF	<R0>

Address	Register name
0x4005_04E0	Reserved
0x4005_04E1	
0x4005_04E2	
0x4005_04E3	
0x4005_04E4	Reserved
0x4005_04E5	
0x4005_04E6	
0x4005_04E7	
0x4005_04E8	Reserved
0x4005_04E9	
0x4005_04EA	
0x4005_04EB	
0x4005_04EC	Reserved
0x4005_04ED	
0x4005_04EE	
0x4005_04EF	

Address	Register name
0x4005_04F0	Reserved
0x4005_04F1	
0x4005_04F2	
0x4005_04F3	
0x4005_04F4	Reserved
0x4005_04F5	
0x4005_04F6	
0x4005_04F7	
0x4005_04F8	Reserved
0x4005_04F9	
0x4005_04FA	
0x4005_04FB	
0x4005_04FC	Reserved
0x4005_04FD	
0x4005_04FE	
0x4005_04FF	

[13] Multi purpose timer (TMR/IGBT) [1/3]

<MPT0>

Address	Register name
0x4005_0800	MT0EN
0x4005_0801	
0x4005_0802	
0x4005_0803	
0x4005_0804	MT0RUN
0x4005_0805	
0x4005_0806	
0x4005_0807	
0x4005_0808	MT0TBCR
0x4005_0809	
0x4005_080A	
0x4005_080B	
0x4005_080C	MT0TBMOD
0x4005_080D	
0x4005_080E	
0x4005_080F	

Address	Register name
0x4005_0810	MT0TBFFCR
0x4005_0811	
0x4005_0812	
0x4005_0813	
0x4005_0814	MT0TBST
0x4005_0815	
0x4005_0816	
0x4005_0817	
0x4005_0818	MT0TBIM
0x4005_0819	
0x4005_081A	
0x4005_081B	
0x4005_081C	MT0TBUC
0x4005_081D	
0x4005_081E	
0x4005_081F	

Address	Register name
0x4005_0820	MT0RG0
0x4005_0821	
0x4005_0822	
0x4005_0823	
0x4005_0824	MT0RG1
0x4005_0825	
0x4005_0826	
0x4005_0827	
0x4005_0828	MT0CP0
0x4005_0829	
0x4005_082A	
0x4005_082B	
0x4005_082C	MT0CP1
0x4005_082D	
0x4005_082E	
0x4005_082F	

Address	Register name
0x4005_0830	MT0IGCR
0x4005_0831	
0x4005_0832	
0x4005_0833	
0x4005_0834	MT0IGRESTA
0x4005_0835	
0x4005_0836	
0x4005_0837	
0x4005_0838	MT0IGST
0x4005_0839	
0x4005_083A	
0x4005_083B	
0x4005_083C	MT0IGICR
0x4005_083D	
0x4005_083E	
0x4005_083F	

Address	Register name
0x4005_0840	MT0IGOCR
0x4005_0841	
0x4005_0842	
0x4005_0843	
0x4005_0844	MT0IGRG2
0x4005_0845	
0x4005_0846	
0x4005_0847	
0x4005_0848	MT0IGRG3
0x4005_0849	
0x4005_084A	
0x4005_084B	
0x4005_084C	MT0IGRG4
0x4005_084D	
0x4005_084E	
0x4005_084F	

Address	Register name
0x4005_0850	MT0IGEMGCR
0x4005_0851	
0x4005_0852	
0x4005_0853	
0x4005_0854	MT0IGEMGST
0x4005_0855	
0x4005_0856	
0x4005_0857	
0x4005_0858	Reserved
0x4005_0859	
0x4005_085A	
0x4005_085B	
0x4005_085C	Reserved
0x4005_085D	
0x4005_085E	
0x4005_085F	

Address	Register name
0x4005_0860	Reserved
0x4005_0861	
0x4005_0862	
0x4005_0863	
0x4005_0864	Reserved
0x4005_0865	
0x4005_0866	
0x4005_0867	
0x4005_0868	Reserved
0x4005_0869	
0x4005_086A	
0x4005_086B	
0x4005_086C	Reserved
0x4005_086D	
0x4005_086E	
0x4005_086F	

Address	Register name
0x4005_0870	Reserved
0x4005_0871	
0x4005_0872	
0x4005_0873	
0x4005_0874	Reserved
0x4005_0875	
0x4005_0876	
0x4005_0877	
0x4005_0878	Reserved
0x4005_0879	
0x4005_087A	
0x4005_087B	
0x4005_087C	Reserved
0x4005_087D	
0x4005_087E	
0x4005_087F	

[13] Multi purpose timer (TMR/IGBT) [2/3]

<MPT1>

Address	Register name
0x4005_0880	MT1EN
0x4005_0881	
0x4005_0882	
0x4005_0883	
0x4005_0884	MT1RUN
0x4005_0885	
0x4005_0886	
0x4005_0887	
0x4005_0888	MT1TBCR
0x4005_0889	
0x4005_088A	
0x4005_088B	
0x4005_088C	MT1TBMOD
0x4005_088D	
0x4005_088E	
0x4005_088F	

Address	Register name
0x4005_0890	MT1TBFFCR
0x4005_0891	
0x4005_0892	
0x4005_0893	
0x4005_0894	MT1TBST
0x4005_0895	
0x4005_0896	
0x4005_0897	
0x4005_0898	MT1TBIM
0x4005_0899	
0x4005_089A	
0x4005_089B	
0x4005_089C	MT11UC
0x4005_089D	
0x4005_089E	
0x4005_089F	

Address	Register name
0x4005_08A0	MT1RG0
0x4005_08A1	
0x4005_08A2	
0x4005_08A3	
0x4005_08A4	MT1RG1
0x4005_08A5	
0x4005_08A6	
0x4005_08A7	
0x4005_08A8	MT1CP0
0x4005_08A9	
0x4005_08AA	
0x4005_08AB	
0x4005_08AC	MT1CP1
0x4005_08AD	
0x4005_08AE	
0x4005_08AF	

Address	Register name
0x4005_08B0	MT1IGCR
0x4005_08B1	
0x4005_08B2	
0x4005_08B3	
0x4005_08B4	MT1IGRESTA
0x4005_08B5	
0x4005_08B6	
0x4005_08B7	
0x4005_08B8	MT1IGST
0x4005_08B9	
0x4005_08BA	
0x4005_08BB	
0x4005_08BC	MT1IGICR
0x4005_08BD	
0x4005_08BE	
0x4005_08BF	

Address	Register name
0x4005_08C0	MT1IGOCR
0x4005_08C1	
0x4005_08C2	
0x4005_08C3	
0x4005_08C4	MT1IGRG2
0x4005_08C5	
0x4005_08C6	
0x4005_08C7	
0x4005_08C8	MT1IGRG3
0x4005_08C9	
0x4005_08CA	
0x4005_08CB	
0x4005_08CC	MT1IGRG4
0x4005_08CD	
0x4005_08CE	
0x4005_08CF	

Address	Register name
0x4005_08D0	MT1IGEMGCR
0x4005_08D1	
0x4005_08D2	
0x4005_08D3	
0x4005_08D4	MT1IGEMGST
0x4005_08D5	
0x4005_08D6	
0x4005_08D7	
0x4005_08D8	Reserved
0x4005_08D9	
0x4005_08DA	
0x4005_08DB	
0x4005_08DC	Reserved
0x4005_08DD	
0x4005_08DE	
0x4005_08DF	

Address	Register name
0x4005_08E0	Reserved
0x4005_08E1	
0x4005_08E2	
0x4005_08E3	
0x4005_08E4	Reserved
0x4005_08E5	
0x4005_08E6	
0x4005_08E7	
0x4005_08E8	Reserved
0x4005_08E9	
0x4005_08EA	
0x4005_08EB	
0x4005_08EC	Reserved
0x4005_08ED	
0x4005_08EE	
0x4005_08EF	

Address	Register name
0x4005_08F0	Reserved
0x4005_08F1	
0x4005_08F2	
0x4005_08F3	
0x4005_08F4	Reserved
0x4005_08F5	
0x4005_08F6	
0x4005_08F7	
0x4005_08F8	Reserved
0x4005_08F9	
0x4005_08FA	
0x4005_08FB	
0x4005_08FC	Reserved
0x4005_08FD	
0x4005_08FE	
0x4005_08FF	

[13] Multi purpose timer (TMR/IGBT) [3/3]

<MPT2>

Address	Register name
0x4005_0900	MT2EN
0x4005_0901	
0x4005_0902	
0x4005_0903	
0x4005_0904	MT2RUN
0x4005_0905	
0x4005_0906	
0x4005_0907	
0x4005_0908	MT2TBCR
0x4005_0909	
0x4005_090A	
0x4005_090B	
0x4005_090C	MT2TBMOD
0x4005_090D	
0x4005_090E	
0x4005_090F	

Address	Register name
0x4005_0910	MT2TBFFCR
0x4005_0911	
0x4005_0912	
0x4005_0913	
0x4005_0914	MT2TBST
0x4005_0915	
0x4005_0916	
0x4005_0917	
0x4005_0918	MT2TBIM
0x4005_0919	
0x4005_091A	
0x4005_091B	
0x4005_091C	MT2TBUC
0x4005_091D	
0x4005_091E	
0x4005_091F	

Address	Register name
0x4005_0920	MT2RG0
0x4005_0921	
0x4005_0922	
0x4005_0923	
0x4005_0924	MT2RG1
0x4005_0925	
0x4005_0926	
0x4005_0927	
0x4005_0928	MT2CP0
0x4005_0929	
0x4005_092A	
0x4005_092B	
0x4005_092C	MT2CP1
0x4005_092D	
0x4005_092E	
0x4005_092F	

Address	Register name
0x4005_0930	MT2IGCR
0x4005_0931	
0x4005_0932	
0x4005_0933	
0x4005_0934	MT2IGRESTA
0x4005_0935	
0x4005_0936	
0x4005_0937	
0x4005_0938	MT2IGST
0x4005_0939	
0x4005_093A	
0x4005_093B	
0x4005_093C	MT2IGICR
0x4005_093D	
0x4005_093E	
0x4005_093F	

Address	Register name
0x4005_0940	MT2IGOCR
0x4005_0941	
0x4005_0942	
0x4005_0943	
0x4005_0944	MT2IGRG2
0x4005_0945	
0x4005_0946	
0x4005_0947	
0x4005_0948	MT2IGRG3
0x4005_0949	
0x4005_094A	
0x4005_094B	
0x4005_094C	MT2IGRG4
0x4005_094D	
0x4005_094E	
0x4005_094F	

Address	Register name
0x4005_0950	MT2IGEMGCR
0x4005_0951	
0x4005_0952	
0x4005_0953	
0x4005_0954	MT2IGEMGST
0x4005_0955	
0x4005_0956	
0x4005_0957	
0x4005_0958	Reserved
0x4005_0959	
0x4005_095A	
0x4005_095B	
0x4005_095C	Reserved
0x4005_095D	
0x4005_095E	
0x4005_095F	

Address	Register name
0x4005_0960	Reserved
0x4005_0961	
0x4005_0962	
0x4005_0963	
0x4005_0964	Reserved
0x4005_0965	
0x4005_0966	
0x4005_0967	
0x4005_0968	Reserved
0x4005_0969	
0x4005_096A	
0x4005_096B	
0x4005_096C	Reserved
0x4005_096D	
0x4005_096E	
0x4005_096F	

Address	Register name
0x4005_0970	Reserved
0x4005_0971	
0x4005_0972	
0x4005_0973	
0x4005_0974	Reserved
0x4005_0975	
0x4005_0976	
0x4005_0977	
0x4005_0978	Reserved
0x4005_0979	
0x4005_097A	
0x4005_097B	
0x4005_097C	Reserved
0x4005_097D	
0x4005_097E	
0x4005_097F	

25.1.14 [15] DMA controller (DMAC)

<DMAC>

Address	Register name
0x4008_0000	DMACInt
0x4008_0001	Status
0x4008_0002	
0x4008_0003	
0x4008_0004	DMACIntTC
0x4008_0005	Status
0x4008_0006	
0x4008_0007	
0x4008_0008	DMACIntTC
0x4008_0009	Clear
0x4008_000A	
0x4008_000B	
0x4008_000C	DMACInt
0x4008_000D	ErrorStatus
0x4008_000E	
0x4008_000F	

Address	Register name
0x4008_0010	DMACInt
0x4008_0011	ErrClr
0x4008_0012	
0x4008_0013	
0x4008_0014	DMACRawInt
0x4008_0015	TCStatus
0x4008_0016	
0x4008_0017	
0x4008_0018	DMACRawInt
0x4008_0019	ErrorStatus
0x4008_001A	
0x4008_001B	
0x4008_001C	DMACEnbld
0x4008_001D	Chns
0x4008_001E	
0x4008_001F	

Address	Register name
0x4008_0020	DMACSoftB
0x4008_0021	Req
0x4008_0022	
0x4008_0023	
0x4008_0024	DMACSoftS
0x4008_0025	Req
0x4008_0026	
0x4008_0027	
0x4008_0028	Reserved
0x4008_0029	
0x4008_002A	
0x4008_002B	
0x4008_002C	Reserved
0x4008_002D	
0x4008_002E	
0x4008_002F	

Address	Register name
0x4008_0030	DMAC
0x4008_0031	Configuration
0x4008_0032	
0x4008_0033	
0x4008_0034	Reserved
0x4008_0035	
0x4008_0036	
0x4008_0037	
0x4008_0038	Reserved
0x4008_0039	
0x4008_003A	
0x4008_003B	
0x4008_003C	Reserved
0x4008_003D	
0x4008_003E	
0x4008_003F	

Address	Register name
0x4008_0100	DMACC0Src
0x4008_0101	Addr
0x4008_0102	
0x4008_0103	
0x4008_0104	DMACC0Dest
0x4008_0105	Addr
0x4008_0106	
0x4008_0107	
0x4008_0108	DMACC0LLI
0x4008_0109	
0x4008_010A	
0x4008_010B	
0x4008_010C	DMACC0
0x4008_010D	Control
0x4008_010E	
0x4008_010F	

Address	Register name
0x4008_0110	DMACC0
0x4008_0111	Configuration
0x4008_0112	
0x4008_0113	
0x4008_0114	Reserved
0x4008_0115	
0x4008_0116	
0x4008_0117	
0x4008_0118	Reserved
0x4008_0119	
0x4008_011A	
0x4008_011B	
0x4008_011C	Reserved
0x4008_011D	
0x4008_011E	
0x4008_011F	

Address	Register name
0x4008_0120	DMACC1Src
0x4008_0121	Addr
0x4008_0122	
0x4008_0123	
0x4008_0124	DMACC1Dest
0x4008_0125	Addr
0x4008_0126	
0x4008_0127	
0x4008_0128	DMACC1LLI
0x4008_0129	
0x4008_012A	
0x4008_012B	
0x4008_012C	DMACC1
0x4008_012D	Control
0x4008_012E	
0x4008_012F	

Address	Register name
0x4008_0130	DMACC1
0x4008_0131	Configuration
0x4008_0132	
0x4008_0133	
0x4008_0134	Reserved
0x4008_0135	
0x4008_0136	
0x4008_0137	
0x4008_0138	Reserved
0x4008_0139	
0x4008_013A	
0x4008_013B	
0x4008_013C	Reserved
0x4008_013D	
0x4008_013E	
0x4008_013F	

25.1.15 [15] SSP controller

<SSP0>

Address	Register name
0x400C_0000	SSP0CR0
0x400C_0001	
0x400C_0002	
0x400C_0003	
0x400C_0004	SSP0CR1
0x400C_0005	
0x400C_0006	
0x400C_0007	
0x400C_0008	SSP0DR
0x400C_0009	
0x400C_000A	
0x400C_000B	
0x400C_000C	SSP0SR
0x400C_000D	
0x400C_000E	
0x400C_000F	

Address	Register name
0x400C_0010	SSP0CPSR
0x400C_0011	
0x400C_0012	
0x400C_0013	
0x400C_0014	SSP0IMSC
0x400C_0015	
0x400C_0016	
0x400C_0017	
0x400C_0018	SSP0RIS
0x400C_0019	
0x400C_001A	
0x400C_001B	
0x400C_001C	SSP0MIS
0x400C_001D	
0x400C_001E	
0x400C_001F	

Address	Register name
0x400C_0020	SSP0ICR
0x400C_0021	
0x400C_0022	
0x400C_0023	
0x400C_0024	SSP0DMACR
0x400C_0025	
0x400C_0026	
0x400C_0027	
0x400C_0028	Reserved
0x400C_0029	
0x400C_002A	
0x400C_002B	
0x400C_002C	Reserved
0x400C_002D	
0x400C_002E	
0x400C_002F	

Address	Register name
0x400C_0030	Reserved
0x400C_0031	
0x400C_0032	
0x400C_0033	
0x400C_0034	Reserved
0x400C_0035	
0x400C_0036	
0x400C_0037	
0x400C_0038	Reserved
0x400C_0039	
0x400C_003A	
0x400C_003B	
0x400C_003C	Reserved
0x400C_003D	
0x400C_003E	
0x400C_003F	

<SSP1>

Address	Register name
0x400C_1000	SSP1CR0
0x400C_1001	
0x400C_1002	
0x400C_1003	
0x400C_1004	SSP1CR1
0x400C_1005	
0x400C_1006	
0x400C_1007	
0x400C_1008	SSP1DR
0x400C_1009	
0x400C_100A	
0x400C_100B	
0x400C_100C	SSP1SR
0x400C_100D	
0x400C_100E	
0x400C_100F	

Address	Register name
0x400C_1010	SSP1CPSR
0x400C_1011	
0x400C_1012	
0x400C_1013	
0x400C_1014	SSP1IMSC
0x400C_1015	
0x400C_1016	
0x400C_1017	
0x400C_1018	SSP1RIS
0x400C_1019	
0x400C_101A	
0x400C_101B	
0x400C_101C	SSP1MIS
0x400C_101D	
0x400C_101E	
0x400C_101F	

Address	Register name
0x400C_1020	SSP1ICR
0x400C_1021	
0x400C_1022	
0x400C_1023	
0x400C_1024	SSP1DMACR
0x400C_1025	
0x400C_1026	
0x400C_1027	
0x400C_1028	Reserved
0x400C_1029	
0x400C_102A	
0x400C_102B	
0x400C_102C	Reserved
0x400C_102D	
0x400C_102E	
0x400C_102F	

Address	Register name
0x400C_1030	Reserved
0x400C_1031	
0x400C_1032	
0x400C_1033	
0x400C_1034	Reserved
0x400C_1035	
0x400C_1036	
0x400C_1037	
0x400C_1038	Reserved
0x400C_1039	
0x400C_103A	
0x400C_103B	
0x400C_103C	Reserved
0x400C_103D	
0x400C_103E	
0x400C_103F	

25.1.16 [16] Flash controller

Address	Register name
0x41FF_F000	Reserved
0x41FF_F001	
0x41FF_F002	
0x41FF_F003	
0x41FF_F004	Reserved
0x41FF_F005	
0x41FF_F006	
0x41FF_F007	
0x41FF_F008	Reserved
0x41FF_F009	
0x41FF_F00A	
0x41FF_F00B	
0x41FF_F00C	Reserved
0x41FF_F00D	
0x41FF_F00E	
0x41FF_F00F	

Address	Register name
0x41FF_F010	FCSECBIT
0x41FF_F011	
0x41FF_F012	
0x41FF_F013	
0x41FF_F014	Reserved
0x41FF_F015	
0x41FF_F016	
0x41FF_F017	
0x41FF_F018	Reserved
0x41FF_F019	
0x41FF_F01A	
0x41FF_F01B	
0x41FF_F01C	Reserved
0x41FF_F01D	
0x41FF_F01E	
0x41FF_F01F	

Address	Register name
0x41FF_F020	FCFLCS
0x41FF_F021	
0x41FF_F022	
0x41FF_F023	
0x41FF_F024	Reserved
0x41FF_F025	
0x41FF_F026	
0x41FF_F027	
0x41FF_F028	Reserved
0x41FF_F029	
0x41FF_F02A	
0x41FF_F02B	
0x41FF_F02C	Reserved
0x41FF_F02D	
0x41FF_F02E	
0x41FF_F02F	

Address	Register name
0x41FF_F030	Reserved
0x41FF_F031	
0x41FF_F032	
0x41FF_F033	
0x41FF_F034	Reserved
0x41FF_F035	
0x41FF_F036	
0x41FF_F037	
0x41FF_F038	Reserved
0x41FF_F039	
0x41FF_F03A	
0x41FF_F03B	
0x41FF_F03C	Reserved
0x41FF_F03D	
0x41FF_F03E	
0x41FF_F03F	

Address	Register name
0x41FF_F040	Reserved
0x41FF_F041	
0x41FF_F042	
0x41FF_F043	
0x41FF_F044	Reserved
0x41FF_F045	
0x41FF_F046	
0x41FF_F047	
0x41FF_F048	Reserved
0x41FF_F049	
0x41FF_F04A	
0x41FF_F04B	
0x41FF_F04C	Reserved
0x41FF_F04D	
0x41FF_F04E	
0x41FF_F04F	

Address	Register name
0x41FF_F050	Reserved
0x41FF_F051	
0x41FF_F052	
0x41FF_F053	
0x41FF_F054	Reserved
0x41FF_F055	
0x41FF_F056	
0x41FF_F057	
0x41FF_F058	Reserved
0x41FF_F059	
0x41FF_F05A	
0x41FF_F05B	
0x41FF_F05C	Reserved
0x41FF_F05D	
0x41FF_F05E	
0x41FF_F05F	

Address	Register name
0x41FF_F060	Reserved
0x41FF_F061	
0x41FF_F062	
0x41FF_F063	
0x41FF_F064	Reserved
0x41FF_F065	
0x41FF_F066	
0x41FF_F067	
0x41FF_F068	Reserved
0x41FF_F069	
0x41FF_F06A	
0x41FF_F06B	
0x41FF_F06C	Reserved
0x41FF_F06D	
0x41FF_F06E	
0x41FF_F06F	

Address	Register name
0x41FF_F070	Reserved
0x41FF_F071	
0x41FF_F072	
0x41FF_F073	
0x41FF_F074	Reserved
0x41FF_F075	
0x41FF_F076	
0x41FF_F077	
0x41FF_F078	Reserved
0x41FF_F079	
0x41FF_F07A	
0x41FF_F07B	
0x41FF_F07C	Reserved
0x41FF_F07D	
0x41FF_F07E	
0x41FF_F07F	

Address	Register name
0x41FF_F080	Reserved
0x41FF_F081	
0x41FF_F082	
0x41FF_F083	
0x41FF_F084	Reserved
0x41FF_F085	
0x41FF_F086	
0x41FF_F087	
0x41FF_F088	Reserved
0x41FF_F089	
0x41FF_F08A	
0x41FF_F08B	
0x41FF_F08C	Reserved
0x41FF_F08D	
0x41FF_F08E	
0x41FF_F08F	

Address	Register name
0x41FF_F090	Reserved
0x41FF_F091	
0x41FF_F092	
0x41FF_F093	
0x41FF_F094	Reserved
0x41FF_F095	
0x41FF_F096	
0x41FF_F097	
0x41FF_F098	Reserved
0x41FF_F099	
0x41FF_F09A	
0x41FF_F09B	
0x41FF_F09C	Reserved
0x41FF_F09D	
0x41FF_F09E	
0x41FF_F09F	

Address	Register name
0x41FF_F0A0	Reserved
0x41FF_F0A1	
0x41FF_F0A2	
0x41FF_F0A3	
0x41FF_F0A4	Reserved
0x41FF_F0A5	
0x41FF_F0A6	
0x41FF_F0A7	
0x41FF_F0A8	Reserved
0x41FF_F0A9	
0x41FF_F0AA	
0x41FF_F0AB	
0x41FF_F0AC	Reserved
0x41FF_F0AD	
0x41FF_F0AE	
0x41FF_F0AF	

Address	Register name
0x41FF_F0B0	Reserved
0x41FF_F0B1	
0x41FF_F0B2	
0x41FF_F0B3	
0x41FF_F0B4	Reserved
0x41FF_F0B5	
0x41FF_F0B6	
0x41FF_F0B7	
0x41FF_F0B8	Reserved
0x41FF_F0B9	
0x41FF_F0BA	
0x41FF_F0BB	
0x41FF_F0BC	Reserved
0x41FF_F0BD	
0x41FF_F0BE	
0x41FF_F0BF	

25.2 Addresses for TMPM382

25.2.1 [1] Port [1/5]

<PORT A>

Address	Register name
0x4000_0000	PADATA
0x4000_0001	<R0>
0x4000_0002	<R0>
0x4000_0003	<R0>
0x4000_0004	PACR
0x4000_0005	<R0>
0x4000_0006	<R0>
0x4000_0007	<R0>
0x4000_0008	PAFR1
0x4000_0009	<R0>
0x4000_000A	<R0>
0x4000_000B	<R0>
0x4000_000C	PAFR2
0x4000_000D	<R0>
0x4000_000E	<R0>
0x4000_000F	<R0>

Address	Register name
0x4000_0010	
0x4000_0011	
0x4000_0012	
0x4000_0013	
0x4000_0014	
0x4000_0015	
0x4000_0016	
0x4000_0017	
0x4000_0018	
0x4000_0019	
0x4000_001A	
0x4000_001B	
0x4000_001C	
0x4000_001D	
0x4000_001E	
0x4000_001F	

Address	Register name
0x4000_0020	
0x4000_0021	
0x4000_0022	
0x4000_0023	
0x4000_0024	
0x4000_0025	
0x4000_0026	
0x4000_0027	
0x4000_0028	PAOD
0x4000_0029	<R0>
0x4000_002A	<R0>
0x4000_002B	<R0>
0x4000_002C	PAPUP
0x4000_002D	<R0>
0x4000_002E	<R0>
0x4000_002F	<R0>

Address	Register name
0x4000_0030	PAPDN
0x4000_0031	<R0>
0x4000_0032	<R0>
0x4000_0033	<R0>
0x4000_0034	
0x4000_0035	
0x4000_0036	
0x4000_0037	
0x4000_0038	PAIE
0x4000_0039	<R0>
0x4000_003A	<R0>
0x4000_003B	<R0>
0x4000_003C	
0x4000_003D	
0x4000_003E	
0x4000_003F	

<PORT B>

Address	Register name
0x4000_0040	PBDATA
0x4000_0041	<R0>
0x4000_0042	<R0>
0x4000_0043	<R0>
0x4000_0044	PBCR
0x4000_0045	<R0>
0x4000_0046	<R0>
0x4000_0047	<R0>
0x4000_0048	PBFR1
0x4000_0049	<R0>
0x4000_004A	<R0>
0x4000_004B	<R0>
0x4000_004C	
0x4000_004D	
0x4000_004E	
0x4000_004F	

Address	Register name
0x4000_0050	
0x4000_0051	
0x4000_0052	
0x4000_0053	
0x4000_0054	
0x4000_0055	
0x4000_0056	
0x4000_0057	
0x4000_0058	
0x4000_0059	
0x4000_005A	
0x4000_005B	
0x4000_005C	
0x4000_005D	
0x4000_005E	
0x4000_005F	

Address	Register name
0x4000_0060	
0x4000_0061	
0x4000_0062	
0x4000_0063	
0x4000_0064	
0x4000_0065	
0x4000_0066	
0x4000_0067	
0x4000_0068	PBOD
0x4000_0069	<R0>
0x4000_006A	<R0>
0x4000_006B	<R0>
0x4000_006C	PBPUP
0x4000_006D	<R0>
0x4000_006E	<R0>
0x4000_006F	<R0>

Address	Register name
0x4000_0070	PBPDN
0x4000_0071	<R0>
0x4000_0072	<R0>
0x4000_0073	<R0>
0x4000_0074	
0x4000_0075	
0x4000_0076	
0x4000_0077	
0x4000_0078	PBIE
0x4000_0079	<R0>
0x4000_007A	<R0>
0x4000_007B	<R0>
0x4000_007C	
0x4000_007D	
0x4000_007E	
0x4000_007F	

<PORT C>

Address	Register name
0x4000_0080	PCDATA
0x4000_0081	<R0>
0x4000_0082	<R0>
0x4000_0083	<R0>
0x4000_0084	PCCR
0x4000_0085	<R0>
0x4000_0086	<R0>
0x4000_0087	<R0>
0x4000_0088	PCFR1
0x4000_0089	<R0>
0x4000_008A	<R0>
0x4000_008B	<R0>
0x4000_008C	PCFR2
0x4000_008D	<R0>
0x4000_008E	<R0>
0x4000_008F	<R0>

Address	Register name
0x4000_0090	PCFR3
0x4000_0091	<R0>
0x4000_0092	<R0>
0x4000_0093	<R0>
0x4000_0094	PCFR4
0x4000_0095	<R0>
0x4000_0096	<R0>
0x4000_0097	<R0>
0x4000_0098	PCFR5
0x4000_0099	<R0>
0x4000_009A	<R0>
0x4000_009B	<R0>
0x4000_009C	
0x4000_009D	
0x4000_009E	
0x4000_009F	

Address	Register name
0x4000_00A0	
0x4000_00A1	
0x4000_00A2	
0x4000_00A3	
0x4000_00A4	
0x4000_00A5	
0x4000_00A6	
0x4000_00A7	
0x4000_00A8	PCOD
0x4000_00A9	<R0>
0x4000_00AA	<R0>
0x4000_00AB	<R0>
0x4000_00AC	PCPUP
0x4000_00AD	<R0>
0x4000_00AE	<R0>
0x4000_00AF	<R0>

Address	Register name
0x4000_00B0	PCPDN
0x4000_00B1	<R0>
0x4000_00B2	<R0>
0x4000_00B3	<R0>
0x4000_00B4	
0x4000_00B5	
0x4000_00B6	
0x4000_00B7	
0x4000_00B8	PCIE
0x4000_00B9	<R0>
0x4000_00BA	<R0>
0x4000_00BB	<R0>
0x4000_00BC	
0x4000_00BD	
0x4000_00BE	
0x4000_00BF	

[1] Port [2/5]

<PORT D>

Address	Register name
0x4000_00C0	Reserved
0x4000_00C1	
0x4000_00C2	
0x4000_00C3	
0x4000_00C4	Reserved
0x4000_00C5	
0x4000_00C6	
0x4000_00C7	
0x4000_00C8	Reserved
0x4000_00C9	
0x4000_00CA	
0x4000_00CB	
0x4000_00CC	Reserved
0x4000_00CD	
0x4000_00CE	
0x4000_00CF	

Address	Register name
0x4000_00D0	Reserved
0x4000_00D1	
0x4000_00D2	
0x4000_00D3	
0x4000_00D4	
0x4000_00D5	
0x4000_00D6	
0x4000_00D7	
0x4000_00D8	
0x4000_00D9	
0x4000_00DA	
0x4000_00DB	
0x4000_00DC	
0x4000_00DD	
0x4000_00DE	
0x4000_00DF	

Address	Register name
0x4000_00E0	
0x4000_00E1	
0x4000_00E2	
0x4000_00E3	
0x4000_00E4	
0x4000_00E5	
0x4000_00E6	
0x4000_00E7	
0x4000_00E8	Reserved
0x4000_00E9	
0x4000_00EA	
0x4000_00EB	
0x4000_00EC	Reserved
0x4000_00ED	
0x4000_00EE	
0x4000_00EF	

Address	Register name
0x4000_00F0	Reserved
0x4000_00F1	
0x4000_00F2	
0x4000_00F3	
0x4000_00F4	
0x4000_00F5	
0x4000_00F6	
0x4000_00F7	
0x4000_00F8	Reserved
0x4000_00F9	
0x4000_00FA	
0x4000_00FB	
0x4000_00FC	
0x4000_00FD	
0x4000_00FE	
0x4000_00FF	

<PORT E>

Address	Register name
0x4000_0100	PEDATA
0x4000_0101	<R0>
0x4000_0102	<R0>
0x4000_0103	<R0>
0x4000_0104	PECR
0x4000_0105	<R0>
0x4000_0106	<R0>
0x4000_0107	<R0>
0x4000_0108	PEFR1
0x4000_0109	<R0>
0x4000_010A	<R0>
0x4000_010B	<R0>
0x4000_010C	PEFR2
0x4000_010D	<R0>
0x4000_010E	<R0>
0x4000_010F	<R0>

Address	Register name
0x4000_0110	
0x4000_0111	
0x4000_0112	
0x4000_0113	
0x4000_0114	
0x4000_0115	
0x4000_0116	
0x4000_0117	
0x4000_0118	
0x4000_0119	
0x4000_011A	
0x4000_011B	
0x4000_011C	
0x4000_011D	
0x4000_011E	
0x4000_011F	

Address	Register name
0x4000_0120	
0x4000_0121	
0x4000_0122	
0x4000_0123	
0x4000_0124	
0x4000_0125	
0x4000_0126	
0x4000_0127	
0x4000_0128	PEOD
0x4000_0129	<R0>
0x4000_012A	<R0>
0x4000_012B	<R0>
0x4000_012C	PEPUP
0x4000_012D	<R0>
0x4000_012E	<R0>
0x4000_012F	<R0>

Address	Register name
0x4000_0130	PEPDN
0x4000_0131	<R0>
0x4000_0132	<R0>
0x4000_0133	<R0>
0x4000_0134	
0x4000_0135	
0x4000_0136	
0x4000_0137	
0x4000_0138	PEIE
0x4000_0139	<R0>
0x4000_013A	<R0>
0x4000_013B	<R0>
0x4000_013C	
0x4000_013D	
0x4000_013E	
0x4000_013F	

<PORT F>

Address	Register name
0x4000_0140	PFDATA
0x4000_0141	<R0>
0x4000_0142	<R0>
0x4000_0143	<R0>
0x4000_0144	PFCR
0x4000_0145	<R0>
0x4000_0146	<R0>
0x4000_0147	<R0>
0x4000_0148	PFFR1
0x4000_0149	<R0>
0x4000_014A	<R0>
0x4000_014B	<R0>
0x4000_014C	PFFR2
0x4000_014D	<R0>
0x4000_014E	<R0>
0x4000_014F	<R0>

Address	Register name
0x4000_0150	PFFR3
0x4000_0151	<R0>
0x4000_0152	<R0>
0x4000_0153	<R0>
0x4000_0154	
0x4000_0155	
0x4000_0156	
0x4000_0157	
0x4000_0158	
0x4000_0159	
0x4000_015A	
0x4000_015B	
0x4000_015C	
0x4000_015D	
0x4000_015E	
0x4000_015F	

Address	Register name
0x4000_0160	
0x4000_0161	
0x4000_0162	
0x4000_0163	
0x4000_0164	
0x4000_0165	
0x4000_0166	
0x4000_0167	
0x4000_0168	PFOD
0x4000_0169	<R0>
0x4000_016A	<R0>
0x4000_016B	<R0>
0x4000_016C	PFPUP
0x4000_016D	<R0>
0x4000_016E	<R0>
0x4000_016F	<R0>

Address	Register name
0x4000_0170	PFPDN
0x4000_0171	<R0>
0x4000_0172	<R0>
0x4000_0173	<R0>
0x4000_0174	
0x4000_0175	
0x4000_0176	
0x4000_0177	
0x4000_0178	PFIE
0x4000_0179	<R0>
0x4000_017A	<R0>
0x4000_017B	<R0>
0x4000_017C	
0x4000_017D	
0x4000_017E	
0x4000_017F	

[1] Port [3/5]

<PORT G>

Address	Register name
0x4000_0180	Reserved
0x4000_0181	
0x4000_0182	
0x4000_0183	
0x4000_0184	Reserved
0x4000_0185	
0x4000_0186	
0x4000_0187	
0x4000_0188	Reserved
0x4000_0189	
0x4000_018A	
0x4000_018B	
0x4000_018C	Reserved
0x4000_018D	
0x4000_018E	
0x4000_018F	

Address	Register name
0x4000_0190	Reserved
0x4000_0191	
0x4000_0192	
0x4000_0193	
0x4000_0194	
0x4000_0195	
0x4000_0196	
0x4000_0197	
0x4000_0198	
0x4000_0199	
0x4000_019A	
0x4000_019B	
0x4000_019C	
0x4000_019D	
0x4000_019E	
0x4000_019F	

Address	Register name
0x4000_01A0	
0x4000_01A1	
0x4000_01A2	
0x4000_01A3	
0x4000_01A4	
0x4000_01A5	
0x4000_01A6	
0x4000_01A7	
0x4000_01A8	Reserved
0x4000_01A9	
0x4000_01AA	
0x4000_01AB	
0x4000_01AC	Reserved
0x4000_01AD	
0x4000_01AE	
0x4000_01AF	

Address	Register name
0x4000_01B0	Reserved
0x4000_01B1	
0x4000_01B2	
0x4000_01B3	
0x4000_01B4	
0x4000_01B5	
0x4000_01B6	
0x4000_01B7	
0x4000_01B8	Reserved
0x4000_01B9	
0x4000_01BA	
0x4000_01BB	
0x4000_01BC	
0x4000_01BD	
0x4000_01BE	
0x4000_01BF	

<PORT H>

Address	Register name
0x4000_01C0	PHDATA
0x4000_01C1	<R0>
0x4000_01C2	<R0>
0x4000_01C3	<R0>
0x4000_01C4	PHCR
0x4000_01C5	<R0>
0x4000_01C6	<R0>
0x4000_01C7	<R0>
0x4000_01C8	PHFR1
0x4000_01C9	<R0>
0x4000_01CA	<R0>
0x4000_01CB	<R0>
0x4000_01CC	
0x4000_01CD	
0x4000_01CE	
0x4000_01CF	

Address	Register name
0x4000_01D0	
0x4000_01D1	
0x4000_01D2	
0x4000_01D3	
0x4000_01D4	
0x4000_01D5	
0x4000_01D6	
0x4000_01D7	
0x4000_01D8	
0x4000_01D9	
0x4000_01DA	
0x4000_01DB	
0x4000_01DC	
0x4000_01DD	
0x4000_01DE	
0x4000_01DF	

Address	Register name
0x4000_01E0	
0x4000_01E1	
0x4000_01E2	
0x4000_01E3	
0x4000_01E4	
0x4000_01E5	
0x4000_01E6	
0x4000_01E7	
0x4000_01E8	PHOD
0x4000_01E9	<R0>
0x4000_01EA	<R0>
0x4000_01EB	<R0>
0x4000_01EC	PHPUP
0x4000_01ED	<R0>
0x4000_01EE	<R0>
0x4000_01EF	<R0>

Address	Register name
0x4000_01F0	PHPDN
0x4000_01F1	<R0>
0x4000_01F2	<R0>
0x4000_01F3	<R0>
0x4000_01F4	
0x4000_01F5	
0x4000_01F6	
0x4000_01F7	
0x4000_01F8	PHIE
0x4000_01F9	<R0>
0x4000_01FA	<R0>
0x4000_01FB	<R0>
0x4000_01FC	
0x4000_01FD	
0x4000_01FE	
0x4000_01FF	

<PORT I>

Address	Register name
0x4000_0200	PIDATA
0x4000_0201	<R0>
0x4000_0202	<R0>
0x4000_0203	<R0>
0x4000_0204	PICR
0x4000_0205	<R0>
0x4000_0206	<R0>
0x4000_0207	<R0>
0x4000_0208	
0x4000_0209	
0x4000_020A	
0x4000_020B	
0x4000_020C	
0x4000_020D	
0x4000_020E	
0x4000_020F	

Address	Register name
0x4000_0210	
0x4000_0211	
0x4000_0212	
0x4000_0213	
0x4000_0214	
0x4000_0215	
0x4000_0216	
0x4000_0217	
0x4000_0218	
0x4000_0219	
0x4000_021A	
0x4000_021B	
0x4000_021C	
0x4000_021D	
0x4000_021E	
0x4000_021F	

Address	Register name
0x4000_0220	
0x4000_0221	
0x4000_0222	
0x4000_0223	
0x4000_0224	
0x4000_0225	
0x4000_0226	
0x4000_0227	
0x4000_0228	PIOD
0x4000_0229	<R0>
0x4000_022A	<R0>
0x4000_022B	<R0>
0x4000_022C	PIPUP
0x4000_022D	<R0>
0x4000_022E	<R0>
0x4000_022F	<R0>

Address	Register name
0x4000_0230	PIPDN
0x4000_0231	<R0>
0x4000_0232	<R0>
0x4000_0233	<R0>
0x4000_0234	
0x4000_0235	
0x4000_0236	
0x4000_0237	
0x4000_0238	PIIE
0x4000_0239	<R0>
0x4000_023A	<R0>
0x4000_023B	<R0>
0x4000_023C	
0x4000_023D	
0x4000_023E	
0x4000_023F	

[1] Port [4/5]

<PORT J>

Address	Register name
0x4000_0240	Reserved
0x4000_0241	
0x4000_0242	
0x4000_0243	
0x4000_0244	Reserved
0x4000_0245	
0x4000_0246	
0x4000_0247	
0x4000_0248	Reserved
0x4000_0249	
0x4000_024A	
0x4000_024B	
0x4000_024C	
0x4000_024D	
0x4000_024E	
0x4000_024F	

Address	Register name
0x4000_0250	
0x4000_0251	
0x4000_0252	
0x4000_0253	
0x4000_0254	
0x4000_0255	
0x4000_0256	
0x4000_0257	
0x4000_0258	
0x4000_0259	
0x4000_025A	
0x4000_025B	
0x4000_025C	
0x4000_025D	
0x4000_025E	
0x4000_025F	

Address	Register name
0x4000_0260	
0x4000_0261	
0x4000_0262	
0x4000_0263	
0x4000_0264	
0x4000_0265	
0x4000_0266	
0x4000_0267	
0x4000_0268	Reserved
0x4000_0269	
0x4000_026A	
0x4000_026B	
0x4000_026C	Reserved
0x4000_026D	
0x4000_026E	
0x4000_026F	

Address	Register name
0x4000_0270	Reserved
0x4000_0271	
0x4000_0272	
0x4000_0273	
0x4000_0274	
0x4000_0275	
0x4000_0276	
0x4000_0277	
0x4000_0278	Reserved
0x4000_0279	
0x4000_027A	
0x4000_027B	
0x4000_027C	
0x4000_027D	
0x4000_027E	
0x4000_027F	

Address	Register name
0x4000_0280	
0x4000_0281	
0x4000_0282	
0x4000_0283	
0x4000_0284	
0x4000_0285	
0x4000_0286	
0x4000_0287	
0x4000_0288	
0x4000_0289	
0x4000_028A	
0x4000_028B	
0x4000_028C	
0x4000_028D	
0x4000_028E	
0x4000_028F	

Address	Register name
0x4000_0290	
0x4000_0291	
0x4000_0292	
0x4000_0293	
0x4000_0294	
0x4000_0295	
0x4000_0296	
0x4000_0297	
0x4000_0298	
0x4000_0299	
0x4000_029A	
0x4000_029B	
0x4000_029C	
0x4000_029D	
0x4000_029E	
0x4000_029F	

Address	Register name
0x4000_02A0	
0x4000_02A1	
0x4000_02A2	
0x4000_02A3	
0x4000_02A4	
0x4000_02A5	
0x4000_02A6	
0x4000_02A7	
0x4000_02A8	
0x4000_02A9	
0x4000_02AA	
0x4000_02AB	
0x4000_02AC	
0x4000_02AD	
0x4000_02AE	
0x4000_02AF	

Address	Register name
0x4000_02B0	
0x4000_02B1	
0x4000_02B2	
0x4000_02B3	
0x4000_02B4	
0x4000_02B5	
0x4000_02B6	
0x4000_02B7	
0x4000_02B8	
0x4000_02B9	
0x4000_02BA	
0x4000_02BB	
0x4000_02BC	
0x4000_02BD	
0x4000_02BE	
0x4000_02BF	

<PORT L>

Address	Register name
0x4000_02C0	PLDATA
0x4000_02C1	<R0>
0x4000_02C2	<R0>
0x4000_02C3	<R0>
0x4000_02C4	PLCR
0x4000_02C5	<R0>
0x4000_02C6	<R0>
0x4000_02C7	<R0>
0x4000_02C8	PLFR1
0x4000_02C9	<R0>
0x4000_02CA	<R0>
0x4000_02CB	<R0>
0x4000_02CC	
0x4000_02CD	
0x4000_02CE	
0x4000_02CF	

Address	Register name
0x4000_02D0	
0x4000_02D1	
0x4000_02D2	
0x4000_02D3	
0x4000_02D4	
0x4000_02D5	
0x4000_02D6	
0x4000_02D7	
0x4000_02D8	
0x4000_02D9	
0x4000_02DA	
0x4000_02DB	
0x4000_02DC	
0x4000_02DD	
0x4000_02DE	
0x4000_02DF	

Address	Register name
0x4000_02E0	
0x4000_02E1	
0x4000_02E2	
0x4000_02E3	
0x4000_02E4	
0x4000_02E5	
0x4000_02E6	
0x4000_02E7	
0x4000_02E8	PLOD
0x4000_02E9	<R0>
0x4000_02EA	<R0>
0x4000_02EB	<R0>
0x4000_02EC	PLPUP
0x4000_02ED	<R0>
0x4000_02EE	<R0>
0x4000_02EF	<R0>

Address	Register name
0x4000_02F0	PLPDN
0x4000_02F1	<R0>
0x4000_02F2	<R0>
0x4000_02F3	<R0>
0x4000_02F4	
0x4000_02F5	
0x4000_02F6	
0x4000_02F7	
0x4000_02F8	PLIE
0x4000_02F9	<R0>
0x4000_02FA	<R0>
0x4000_02FB	<R0>
0x4000_02FC	
0x4000_02FD	
0x4000_02FE	
0x4000_02FF	

[1] Port [5/5]

<PORT M>

Address	Register name
0x4000_0300	PMDATA
0x4000_0301	<R0>
0x4000_0302	<R0>
0x4000_0303	<R0>
0x4000_0304	PMCR
0x4000_0305	<R0>
0x4000_0306	<R0>
0x4000_0307	<R0>
0x4000_0308	
0x4000_0309	
0x4000_030A	
0x4000_030B	
0x4000_030C	
0x4000_030D	
0x4000_030E	
0x4000_030F	

Address	Register name
0x4000_0310	
0x4000_0311	
0x4000_0312	
0x4000_0313	
0x4000_0314	
0x4000_0315	
0x4000_0316	
0x4000_0317	
0x4000_0318	
0x4000_0319	
0x4000_031A	
0x4000_031B	
0x4000_031C	
0x4000_031D	
0x4000_031E	
0x4000_031F	

Address	Register name
0x4000_0320	
0x4000_0321	
0x4000_0322	
0x4000_0323	
0x4000_0324	
0x4000_0325	
0x4000_0326	
0x4000_0327	
0x4000_0328	PMOD
0x4000_0329	<R0>
0x4000_032A	<R0>
0x4000_032B	<R0>
0x4000_032C	PMPUP
0x4000_032D	<R0>
0x4000_032E	<R0>
0x4000_032F	<R0>

Address	Register name
0x4000_0330	PMPDN
0x4000_0331	<R0>
0x4000_0332	<R0>
0x4000_0333	<R0>
0x4000_0334	
0x4000_0335	
0x4000_0336	
0x4000_0337	
0x4000_0338	PMIE
0x4000_0339	<R0>
0x4000_033A	<R0>
0x4000_033B	<R0>
0x4000_033C	
0x4000_033D	
0x4000_033E	
0x4000_033F	

<PORT N>

Address	Register name
0x4000_0340	
0x4000_0341	
0x4000_0342	
0x4000_0343	
0x4000_0344	
0x4000_0345	
0x4000_0346	
0x4000_0347	
0x4000_0348	
0x4000_0349	
0x4000_034A	
0x4000_034B	
0x4000_034C	
0x4000_034D	
0x4000_034E	
0x4000_034F	

Address	Register name
0x4000_0350	
0x4000_0351	
0x4000_0352	
0x4000_0353	
0x4000_0354	
0x4000_0355	
0x4000_0356	
0x4000_0357	
0x4000_0358	
0x4000_0359	
0x4000_035A	
0x4000_035B	
0x4000_035C	
0x4000_035D	
0x4000_035E	
0x4000_035F	

Address	Register name
0x4000_0360	
0x4000_0361	
0x4000_0362	
0x4000_0363	
0x4000_0364	
0x4000_0365	
0x4000_0366	
0x4000_0367	
0x4000_0368	
0x4000_0369	
0x4000_036A	
0x4000_036B	
0x4000_036C	
0x4000_036D	
0x4000_036E	
0x4000_036F	

Address	Register name
0x4000_0370	
0x4000_0371	
0x4000_0372	
0x4000_0373	
0x4000_0374	
0x4000_0375	
0x4000_0376	
0x4000_0377	
0x4000_0378	
0x4000_0379	
0x4000_037A	
0x4000_037B	
0x4000_037C	
0x4000_037D	
0x4000_037E	
0x4000_037F	

<PORT P>

Address	Register name
0x4000_0380	PPDATA
0x4000_0381	<R0>
0x4000_0382	<R0>
0x4000_0383	<R0>
0x4000_0384	PPCR
0x4000_0385	<R0>
0x4000_0386	<R0>
0x4000_0387	<R0>
0x4000_0388	
0x4000_0389	
0x4000_038A	
0x4000_038B	
0x4000_038C	
0x4000_038D	
0x4000_038E	
0x4000_038F	

Address	Register name
0x4000_0390	
0x4000_0391	
0x4000_0392	
0x4000_0393	
0x4000_0394	
0x4000_0395	
0x4000_0396	
0x4000_0397	
0x4000_0398	
0x4000_0399	
0x4000_039A	
0x4000_039B	
0x4000_039C	
0x4000_039D	
0x4000_039E	
0x4000_039F	

Address	Register name
0x4000_03A0	
0x4000_03A1	
0x4000_03A2	
0x4000_03A3	
0x4000_03A4	
0x4000_03A5	
0x4000_03A6	
0x4000_03A7	
0x4000_03A8	PPOD
0x4000_03A9	<R0>
0x4000_03AA	<R0>
0x4000_03AB	<R0>
0x4000_03AC	PPPUP
0x4000_03AD	<R0>
0x4000_03AE	<R0>
0x4000_03AF	<R0>

Address	Register name
0x4000_03B0	PPPDN
0x4000_03B1	<R0>
0x4000_03B2	<R0>
0x4000_03B3	<R0>
0x4000_03B4	
0x4000_03B5	
0x4000_03B6	
0x4000_03B7	
0x4000_03B8	PPIE
0x4000_03B9	<R0>
0x4000_03BA	<R0>
0x4000_03BB	<R0>
0x4000_03BC	
0x4000_03BD	
0x4000_03BE	
0x4000_03BF	

25.2.2 [2] 16-bit timer [1/3]

<TMRB0>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0000	TB0EN	0x4001_0010	TB0FFCR	0x4001_0020	TB0RG0	0x4001_0030	
0x4001_0001	<R0>	0x4001_0011	<R0>	0x4001_0021		0x4001_0031	
0x4001_0002	<R0>	0x4001_0012	<R0>	0x4001_0022	<R0>	0x4001_0032	
0x4001_0003	<R0>	0x4001_0013	<R0>	0x4001_0023	<R0>	0x4001_0033	
0x4001_0004	TB0RUN	0x4001_0014	TB0ST	0x4001_0024	TB0RG1	0x4001_0034	
0x4001_0005	<R0>	0x4001_0015	<R0>	0x4001_0025		0x4001_0035	
0x4001_0006	<R0>	0x4001_0016	<R0>	0x4001_0026	<R0>	0x4001_0036	
0x4001_0007	<R0>	0x4001_0017	<R0>	0x4001_0027	<R0>	0x4001_0037	
0x4001_0008	TB0CR	0x4001_0018	TB0IM	0x4001_0028	TB0CP0	0x4001_0038	
0x4001_0009	<R0>	0x4001_0019	<R0>	0x4001_0029		0x4001_0039	
0x4001_000A	<R0>	0x4001_001A	<R0>	0x4001_002A	<R0>	0x4001_003A	
0x4001_000B	<R0>	0x4001_001B	<R0>	0x4001_002B	<R0>	0x4001_003B	
0x4001_000C	TB0MOD	0x4001_001C	TB0UC	0x4001_002C	TB0CP1	0x4001_003C	
0x4001_000D	<R0>	0x4001_001D		0x4001_002D		0x4001_003D	
0x4001_000E	<R0>	0x4001_001E	<R0>	0x4001_002E	<R0>	0x4001_003E	
0x4001_000F	<R0>	0x4001_001F	<R0>	0x4001_002F	<R0>	0x4001_003F	

<TMRB1>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0040	TB1EN	0x4001_0050	TB1FFCR	0x4001_0060	TB1RG0	0x4001_0070	
0x4001_0041	<R0>	0x4001_0051	<R0>	0x4001_0061		0x4001_0071	
0x4001_0042	<R0>	0x4001_0052	<R0>	0x4001_0062	<R0>	0x4001_0072	
0x4001_0043	<R0>	0x4001_0053	<R0>	0x4001_0063	<R0>	0x4001_0073	
0x4001_0044	TB1RUN	0x4001_0054	TB1ST	0x4001_0064	TB1RG1	0x4001_0074	
0x4001_0045	<R0>	0x4001_0055	<R0>	0x4001_0065		0x4001_0075	
0x4001_0046	<R0>	0x4001_0056	<R0>	0x4001_0066	<R0>	0x4001_0076	
0x4001_0047	<R0>	0x4001_0057	<R0>	0x4001_0067	<R0>	0x4001_0077	
0x4001_0048	TB1CR	0x4001_0058	TB1IM	0x4001_0068	TB1CP0	0x4001_0078	
0x4001_0049	<R0>	0x4001_0059	<R0>	0x4001_0069		0x4001_0079	
0x4001_004A	<R0>	0x4001_005A	<R0>	0x4001_006A	<R0>	0x4001_007A	
0x4001_004B	<R0>	0x4001_005B	<R0>	0x4001_006B	<R0>	0x4001_007B	
0x4001_004C	TB1MOD	0x4001_005C	TB1UC	0x4001_006C	TB1CP1	0x4001_007C	
0x4001_004D	<R0>	0x4001_005D		0x4001_006D		0x4001_007D	
0x4001_004E	<R0>	0x4001_005E	<R0>	0x4001_006E	<R0>	0x4001_007E	
0x4001_004F	<R0>	0x4001_005F	<R0>	0x4001_006F	<R0>	0x4001_007F	

<TMRB2>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0080	TB2EN	0x4001_0090	TB2FFCR	0x4001_00A0	TB2RG0	0x4001_00B0	
0x4001_0081	<R0>	0x4001_0091	<R0>	0x4001_00A1		0x4001_00B1	
0x4001_0082	<R0>	0x4001_0092	<R0>	0x4001_00A2	<R0>	0x4001_00B2	
0x4001_0083	<R0>	0x4001_0093	<R0>	0x4001_00A3	<R0>	0x4001_00B3	
0x4001_0084	TB2RUN	0x4001_0094	TB2ST	0x4001_00A4	TB2RG1	0x4001_00B4	
0x4001_0085	<R0>	0x4001_0095	<R0>	0x4001_00A5		0x4001_00B5	
0x4001_0086	<R0>	0x4001_0096	<R0>	0x4001_00A6	<R0>	0x4001_00B6	
0x4001_0087	<R0>	0x4001_0097	<R0>	0x4001_00A7	<R0>	0x4001_00B7	
0x4001_0088	TB2CR	0x4001_0098	TB2IM	0x4001_00A8	TB2CP0	0x4001_00B8	
0x4001_0089	<R0>	0x4001_0099	<R0>	0x4001_00A9		0x4001_00B9	
0x4001_008A	<R0>	0x4001_009A	<R0>	0x4001_00AA	<R0>	0x4001_00BA	
0x4001_008B	<R0>	0x4001_009B	<R0>	0x4001_00AB	<R0>	0x4001_00BB	
0x4001_008C	TB2MOD	0x4001_009C	TB2UC	0x4001_00AC	TB2CP1	0x4001_00BC	
0x4001_008D	<R0>	0x4001_009D		0x4001_00AD		0x4001_00BD	
0x4001_008E	<R0>	0x4001_009E	<R0>	0x4001_00AE	<R0>	0x4001_00BE	
0x4001_008F	<R0>	0x4001_009F	<R0>	0x4001_00AF	<R0>	0x4001_00BF	

[2] 16-bit timer [2/3]

<TMRB3>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_00C0	TB3EN	0x4001_00D0	Reserved	0x4001_00E0	TB3RG0	0x4001_00F0	
0x4001_00C1	<R0>	0x4001_00D1		0x4001_00E1		0x4001_00F1	
0x4001_00C2	<R0>	0x4001_00D2		0x4001_00E2	<R0>	0x4001_00F2	
0x4001_00C3	<R0>	0x4001_00D3		0x4001_00E3	<R0>	0x4001_00F3	
0x4001_00C4	TB3RUN	0x4001_00D4	TB3ST	0x4001_00E4	TB3RG1	0x4001_00F4	
0x4001_00C5	<R0>	0x4001_00D5	<R0>	0x4001_00E5		0x4001_00F5	
0x4001_00C6	<R0>	0x4001_00D6	<R0>	0x4001_00E6	<R0>	0x4001_00F6	
0x4001_00C7	<R0>	0x4001_00D7	<R0>	0x4001_00E7	<R0>	0x4001_00F7	
0x4001_00C8	TB3CR	0x4001_00D8	TB3IM	0x4001_00E8	TB3CP0	0x4001_00F8	
0x4001_00C9	<R0>	0x4001_00D9	<R0>	0x4001_00E9		0x4001_00F9	
0x4001_00CA	<R0>	0x4001_00DA	<R0>	0x4001_00EA	<R0>	0x4001_00FA	
0x4001_00CB	<R0>	0x4001_00DB	<R0>	0x4001_00EB	<R0>	0x4001_00FB	
0x4001_00CC	TB3MOD	0x4001_00DC	TB3UC	0x4001_00EC	TB3CP1	0x4001_00FC	
0x4001_00CD	<R0>	0x4001_00DD		0x4001_00ED		0x4001_00FD	
0x4001_00CE	<R0>	0x4001_00DE	<R0>	0x4001_00EE	<R0>	0x4001_00FE	
0x4001_00CF	<R0>	0x4001_00DF	<R0>	0x4001_00EF	<R0>	0x4001_00FF	

<TMRB4>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0100	TB4EN	0x4001_0110	TB4FFCR	0x4001_0120	TB4RG0	0x4001_0130	
0x4001_0101	<R0>	0x4001_0111	<R0>	0x4001_0121		0x4001_0131	
0x4001_0102	<R0>	0x4001_0112	<R0>	0x4001_0122	<R0>	0x4001_0132	
0x4001_0103	<R0>	0x4001_0113	<R0>	0x4001_0123	<R0>	0x4001_0133	
0x4001_0104	TB4RUN	0x4001_0114	TB4ST	0x4001_0124	TB4RG1	0x4001_0134	
0x4001_0105	<R0>	0x4001_0115	<R0>	0x4001_0125		0x4001_0135	
0x4001_0106	<R0>	0x4001_0116	<R0>	0x4001_0126	<R0>	0x4001_0136	
0x4001_0107	<R0>	0x4001_0117	<R0>	0x4001_0127	<R0>	0x4001_0137	
0x4001_0108	TB4CR	0x4001_0118	TB4IM	0x4001_0128	TB4CP0	0x4001_0138	
0x4001_0109	<R0>	0x4001_0119	<R0>	0x4001_0129		0x4001_0139	
0x4001_010A	<R0>	0x4001_011A	<R0>	0x4001_012A	<R0>	0x4001_013A	
0x4001_010B	<R0>	0x4001_011B	<R0>	0x4001_012B	<R0>	0x4001_013B	
0x4001_010C	TB4MOD	0x4001_011C	TB4UC	0x4001_012C	TB4CP1	0x4001_013C	
0x4001_010D	<R0>	0x4001_011D		0x4001_012D		0x4001_013D	
0x4001_010E	<R0>	0x4001_011E	<R0>	0x4001_012E	<R0>	0x4001_013E	
0x4001_010F	<R0>	0x4001_011F	<R0>	0x4001_012F	<R0>	0x4001_013F	

<TMRB5>

Address	Register name	Address	Register name	Address	Register name	Address	Register name
0x4001_0140	TB5EN	0x4001_0150	TB5FFCR	0x4001_0160	TB5RG0	0x4001_0170	
0x4001_0141	<R0>	0x4001_0151	<R0>	0x4001_0161		0x4001_0171	
0x4001_0142	<R0>	0x4001_0152	<R0>	0x4001_0162	<R0>	0x4001_0172	
0x4001_0143	<R0>	0x4001_0153	<R0>	0x4001_0163	<R0>	0x4001_0173	
0x4001_0144	TB5RUN	0x4001_0154	TB5ST	0x4001_0164	TB5RG1	0x4001_0174	
0x4001_0145	<R0>	0x4001_0155	<R0>	0x4001_0165		0x4001_0175	
0x4001_0146	<R0>	0x4001_0156	<R0>	0x4001_0166	<R0>	0x4001_0176	
0x4001_0147	<R0>	0x4001_0157	<R0>	0x4001_0167	<R0>	0x4001_0177	
0x4001_0148	TB5CR	0x4001_0158	TB5IM	0x4001_0168	TB5CP0	0x4001_0178	
0x4001_0149	<R0>	0x4001_0159	<R0>	0x4001_0169		0x4001_0179	
0x4001_014A	<R0>	0x4001_015A	<R0>	0x4001_016A	<R0>	0x4001_017A	
0x4001_014B	<R0>	0x4001_015B	<R0>	0x4001_016B	<R0>	0x4001_017B	
0x4001_014C	TB5MOD	0x4001_015C	TB5UC	0x4001_016C	TB5CP1	0x4001_017C	
0x4001_014D	<R0>	0x4001_015D		0x4001_016D		0x4001_017D	
0x4001_014E	<R0>	0x4001_015E	<R0>	0x4001_016E	<R0>	0x4001_017E	
0x4001_014F	<R0>	0x4001_015F	<R0>	0x4001_016F	<R0>	0x4001_017F	

[2] 16-bit timer [3/3]

<TMRB6>

Address	Register name
0x4001_0180	TB6EN
0x4001_0181	<R0>
0x4001_0182	<R0>
0x4001_0183	<R0>
0x4001_0184	TB6RUN
0x4001_0185	<R0>
0x4001_0186	<R0>
0x4001_0187	<R0>
0x4001_0188	TB6CR
0x4001_0189	<R0>
0x4001_018A	<R0>
0x4001_018B	<R0>
0x4001_018C	TB6MOD
0x4001_018D	<R0>
0x4001_018E	<R0>
0x4001_018F	<R0>

Address	Register name
0x4001_0190	TB6FFCR
0x4001_0191	<R0>
0x4001_0192	<R0>
0x4001_0193	<R0>
0x4001_0194	TB6ST
0x4001_0195	<R0>
0x4001_0196	<R0>
0x4001_0197	<R0>
0x4001_0198	TB6IM
0x4001_0199	<R0>
0x4001_019A	<R0>
0x4001_019B	<R0>
0x4001_019C	TB6UC
0x4001_019D	
0x4001_019E	<R0>
0x4001_019F	<R0>

Address	Register name
0x4001_01A0	TB6RG0
0x4001_01A1	
0x4001_01A2	<R0>
0x4001_01A3	<R0>
0x4001_01A4	TB6RG1
0x4001_01A5	
0x4001_01A6	<R0>
0x4001_01A7	<R0>
0x4001_01A8	TB6CP0
0x4001_01A9	
0x4001_01AA	<R0>
0x4001_01AB	<R0>
0x4001_01AC	TB6CP1
0x4001_01AD	
0x4001_01AE	<R0>
0x4001_01AF	<R0>

Address	Register name
0x4001_01B0	
0x4001_01B1	
0x4001_01B2	
0x4001_01B3	
0x4001_01B4	
0x4001_01B5	
0x4001_01B6	
0x4001_01B7	
0x4001_01B8	
0x4001_01B9	
0x4001_01BA	
0x4001_01BB	
0x4001_01BC	
0x4001_01BD	
0x4001_01BE	
0x4001_01BF	

<TMRB7>

Address	Register name
0x4001_01C0	TB7EN
0x4001_01C1	<R0>
0x4001_01C2	<R0>
0x4001_01C3	<R0>
0x4001_01C4	TB7RUN
0x4001_01C5	<R0>
0x4001_01C6	<R0>
0x4001_01C7	<R0>
0x4001_01C8	TB7CR
0x4001_01C9	<R0>
0x4001_01CA	<R0>
0x4001_01CB	<R0>
0x4001_01CC	TB7MOD
0x4001_01CD	<R0>
0x4001_01CE	<R0>
0x4001_01CF	<R0>

Address	Register name
0x4001_01D0	TB7FFCR
0x4001_01D1	<R0>
0x4001_01D2	<R0>
0x4001_01D3	<R0>
0x4001_01D4	TB7ST
0x4001_01D5	<R0>
0x4001_01D6	<R0>
0x4001_01D7	<R0>
0x4001_01D8	TB7IM
0x4001_01D9	<R0>
0x4001_01DA	<R0>
0x4001_01DB	<R0>
0x4001_01DC	TB7UC
0x4001_01DD	
0x4001_01DE	<R0>
0x4001_01DF	<R0>

Address	Register name
0x4001_01E0	TB7RG0
0x4001_01E1	
0x4001_01E2	<R0>
0x4001_01E3	<R0>
0x4001_01E4	TB7RG1
0x4001_01E5	
0x4001_01E6	<R0>
0x4001_01E7	<R0>
0x4001_01E8	TB7CP0
0x4001_01E9	
0x4001_01EA	<R0>
0x4001_01EB	<R0>
0x4001_01EC	TB7CP1
0x4001_01ED	
0x4001_01EE	<R0>
0x4001_01EF	<R0>

Address	Register name
0x4001_01F0	
0x4001_01F1	
0x4001_01F2	
0x4001_01F3	
0x4001_01F4	
0x4001_01F5	
0x4001_01F6	
0x4001_01F7	
0x4001_01F8	
0x4001_01F9	
0x4001_01FA	
0x4001_01FB	
0x4001_01FC	
0x4001_01FD	
0x4001_01FE	
0x4001_01FF	

25.2.3 [3] Encoder input (ENC)

<ENC0>

Address	Register name
0x4001_0400	Reserved
0x4001_0401	
0x4001_0402	
0x4001_0403	
0x4001_0404	Reserved
0x4001_0405	
0x4001_0406	
0x4001_0407	
0x4001_0408	Reserved
0x4001_0409	
0x4001_040A	
0x4001_040B	
0x4001_040C	Reserved
0x4001_040D	
0x4001_040E	
0x4001_040F	

Address	Register name
0x4001_0410	Reserved
0x4001_0411	
0x4001_0412	
0x4001_0413	
0x4001_0414	Reserved
0x4001_0415	
0x4001_0416	
0x4001_0417	
0x4001_0418	Reserved
0x4001_0419	
0x4001_041A	
0x4001_041B	
0x4001_041C	Reserved
0x4001_041D	
0x4001_041E	
0x4001_041F	

Address	Register name
0x4001_0420	Reserved
0x4001_0421	
0x4001_0422	
0x4001_0423	
0x4001_0424	Reserved
0x4001_0425	
0x4001_0426	
0x4001_0427	
0x4001_0428	Reserved
0x4001_0429	
0x4001_042A	
0x4001_042B	
0x4001_042C	Reserved
0x4001_042D	
0x4001_042E	
0x4001_042F	

Address	Register name
0x4001_0430	Reserved
0x4001_0431	
0x4001_0432	
0x4001_0433	
0x4001_0434	Reserved
0x4001_0435	
0x4001_0436	
0x4001_0437	
0x4001_0438	Reserved
0x4001_0439	
0x4001_043A	
0x4001_043B	
0x4001_043C	Reserved
0x4001_043D	
0x4001_043E	
0x4001_043F	

<ENC1>

Address	Register name
0x4001_0500	Reserved
0x4001_0501	
0x4001_0502	
0x4001_0503	
0x4001_0504	Reserved
0x4001_0505	
0x4001_0506	
0x4001_0507	
0x4001_0508	Reserved
0x4001_0509	
0x4001_050A	
0x4001_050B	
0x4001_050C	Reserved
0x4001_050D	
0x4001_050E	
0x4001_050F	

Address	Register name
0x4001_0510	Reserved
0x4001_0511	
0x4001_0512	
0x4001_0513	
0x4001_0514	Reserved
0x4001_0515	
0x4001_0516	
0x4001_0517	
0x4001_0518	Reserved
0x4001_0519	
0x4001_051A	
0x4001_051B	
0x4001_051C	Reserved
0x4001_051D	
0x4001_051E	
0x4001_051F	

Address	Register name
0x4001_0520	Reserved
0x4001_0521	
0x4001_0522	
0x4001_0523	
0x4001_0524	Reserved
0x4001_0525	
0x4001_0526	
0x4001_0527	
0x4001_0528	Reserved
0x4001_0529	
0x4001_052A	
0x4001_052B	
0x4001_052C	Reserved
0x4001_052D	
0x4001_052E	
0x4001_052F	

Address	Register name
0x4001_0530	Reserved
0x4001_0531	
0x4001_0532	
0x4001_0533	
0x4001_0534	Reserved
0x4001_0535	
0x4001_0536	
0x4001_0537	
0x4001_0538	Reserved
0x4001_0539	
0x4001_053A	
0x4001_053B	
0x4001_053C	Reserved
0x4001_053D	
0x4001_053E	
0x4001_053F	

25.2.4 [4] Serial bus interface (SBI)

<SBI0>

Address	Register name
0x4002_0000	SBI0CR0
0x4002_0001	
0x4002_0002	
0x4002_0003	
0x4002_0004	SBI0CR1
0x4002_0005	
0x4002_0006	
0x4002_0007	
0x4002_0008	SBI0DBR
0x4002_0009	
0x4002_000A	
0x4002_000B	
0x4002_000C	SBI0I2CAR
0x4002_000D	
0x4002_000E	
0x4002_000F	

Address	Register name
0x4002_0010	SBI0CR2/SR
0x4002_0011	
0x4002_0012	
0x4002_0013	
0x4002_0014	SBI0BR0
0x4002_0015	
0x4002_0016	
0x4002_0017	
0x4002_0018	
0x4002_0019	
0x4002_001A	
0x4002_001B	
0x4002_001C	
0x4002_001D	
0x4002_001E	
0x4002_001F	

Address	Register name
0x4002_0020	Reserved
0x4002_0021	
0x4002_0022	
0x4002_0023	
0x4002_0024	Reserved
0x4002_0025	
0x4002_0026	
0x4002_0027	
0x4002_0028	Reserved
0x4002_0029	
0x4002_002A	
0x4002_002B	
0x4002_002C	Reserved
0x4002_002D	
0x4002_002E	
0x4002_002F	

Address	Register name
0x4002_0030	Reserved
0x4002_0031	
0x4002_0032	
0x4002_0033	
0x4002_0034	Reserved
0x4002_0035	
0x4002_0036	
0x4002_0037	
0x4002_0038	
0x4002_0039	
0x4002_003A	
0x4002_003B	
0x4002_003C	
0x4002_003D	
0x4002_003E	
0x4002_003F	

Address	Register name
0x4002_0040	
0x4002_0041	
0x4002_0042	
0x4002_0043	
0x4002_0044	
0x4002_0045	
0x4002_0046	
0x4002_0047	
0x4002_0048	
0x4002_0049	
0x4002_004A	
0x4002_004B	
0x4002_004C	
0x4002_004D	
0x4002_004E	
0x4002_004F	

Address	Register name
0x4002_0050	
0x4002_0051	
0x4002_0052	
0x4002_0053	
0x4002_0054	
0x4002_0055	
0x4002_0056	
0x4002_0057	
0x4002_0058	
0x4002_0059	
0x4002_005A	
0x4002_005B	
0x4002_005C	
0x4002_005D	
0x4002_005E	
0x4002_005F	

Address	Register name
0x4002_0060	
0x4002_0061	
0x4002_0062	
0x4002_0063	
0x4002_0064	
0x4002_0065	
0x4002_0066	
0x4002_0067	
0x4002_0068	
0x4002_0069	
0x4002_006A	
0x4002_006B	
0x4002_006C	
0x4002_006D	
0x4002_006E	
0x4002_006F	

Address	Register name
0x4002_0070	
0x4002_0071	
0x4002_0072	
0x4002_0073	
0x4002_0074	
0x4002_0075	
0x4002_0076	
0x4002_0077	
0x4002_0078	
0x4002_0079	
0x4002_007A	
0x4002_007B	
0x4002_007C	
0x4002_007D	
0x4002_007E	
0x4002_007F	

25.2.5 [5] Serial interface (UART/SIO)

<SIO0>

Address	Register name
0x4002_0080	SC0EN
0x4002_0081	<R0>
0x4002_0082	<R0>
0x4002_0083	<R0>
0x4002_0084	SC0BUF
0x4002_0085	<R0>
0x4002_0086	<R0>
0x4002_0087	<R0>
0x4002_0088	SC0CR
0x4002_0089	<R0>
0x4002_008A	<R0>
0x4002_008B	<R0>
0x4002_008C	SC0MOD0
0x4002_008D	<R0>
0x4002_008E	<R0>
0x4002_008F	<R0>

Address	Register name
0x4002_0090	SC0BRCR
0x4002_0091	<R0>
0x4002_0092	<R0>
0x4002_0093	<R0>
0x4002_0094	SC0BRADD
0x4002_0095	<R0>
0x4002_0096	<R0>
0x4002_0097	<R0>
0x4002_0098	SC0MOD1
0x4002_0099	<R0>
0x4002_009A	<R0>
0x4002_009B	<R0>
0x4002_009C	SC0MOD2
0x4002_009D	<R0>
0x4002_009E	<R0>
0x4002_009F	<R0>

Address	Register name
0x4002_00A0	SC0RFC
0x4002_00A1	<R0>
0x4002_00A2	<R0>
0x4002_00A3	<R0>
0x4002_00A4	SC0TFC
0x4002_00A5	<R0>
0x4002_00A6	<R0>
0x4002_00A7	<R0>
0x4002_00A8	SC0RST
0x4002_00A9	<R0>
0x4002_00AA	<R0>
0x4002_00AB	<R0>
0x4002_00AC	SC0TST
0x4002_00AD	<R0>
0x4002_00AE	<R0>
0x4002_00AF	<R0>

Address	Register name
0x4002_00B0	SC0FCNF
0x4002_00B1	<R0>
0x4002_00B2	<R0>
0x4002_00B3	<R0>
0x4002_00B4	
0x4002_00B5	
0x4002_00B6	
0x4002_00B7	
0x4002_00B8	
0x4002_00B9	
0x4002_00BA	
0x4002_00BB	
0x4002_00BC	
0x4002_00BD	
0x4002_00BE	
0x4002_00BF	

<SIO1>

Address	Register name
0x4002_00C0	SC1EN
0x4002_00C1	<R0>
0x4002_00C2	<R0>
0x4002_00C3	<R0>
0x4002_00C4	SC1BUF
0x4002_00C5	<R0>
0x4002_00C6	<R0>
0x4002_00C7	<R0>
0x4002_00C8	SC1CR
0x4002_00C9	<R0>
0x4002_00CA	<R0>
0x4002_00CB	<R0>
0x4002_00CC	SC1MOD0
0x4002_00CD	<R0>
0x4002_00CE	<R0>
0x4002_00CF	<R0>

Address	Register name
0x4002_00D0	SC1BRCR
0x4002_00D1	<R0>
0x4002_00D2	<R0>
0x4002_00D3	<R0>
0x4002_00D4	SC1BRADD
0x4002_00D5	<R0>
0x4002_00D6	<R0>
0x4002_00D7	<R0>
0x4002_00D8	SC1MOD1
0x4002_00D9	<R0>
0x4002_00DA	<R0>
0x4002_00DB	<R0>
0x4002_00DC	SC1MOD2
0x4002_00DD	<R0>
0x4002_00DE	<R0>
0x4002_00DF	<R0>

Address	Register name
0x4002_00E0	SC1RFC
0x4002_00E1	<R0>
0x4002_00E2	<R0>
0x4002_00E3	<R0>
0x4002_00E4	SC1TFC
0x4002_00E5	<R0>
0x4002_00E6	<R0>
0x4002_00E7	<R0>
0x4002_00E8	SC1RST
0x4002_00E9	<R0>
0x4002_00EA	<R0>
0x4002_00EB	<R0>
0x4002_00EC	SC1TST
0x4002_00ED	<R0>
0x4002_00EE	<R0>
0x4002_00EF	<R0>

Address	Register name
0x4002_00F0	SC1FCNF
0x4002_00F1	<R0>
0x4002_00F2	<R0>
0x4002_00F3	<R0>
0x4002_00F4	
0x4002_00F5	
0x4002_00F6	
0x4002_00F7	
0x4002_00F8	
0x4002_00F9	
0x4002_00FA	
0x4002_00FB	
0x4002_00FC	
0x4002_00FD	
0x4002_00FE	
0x4002_00FF	

<SIO2>

Address	Register name
0x4002_0100	Reserved
0x4002_0101	
0x4002_0102	
0x4002_0103	
0x4002_0104	Reserved
0x4002_0105	
0x4002_0106	
0x4002_0107	
0x4002_0108	Reserved
0x4002_0109	
0x4002_010A	
0x4002_010B	
0x4002_010C	Reserved
0x4002_010D	
0x4002_010E	
0x4002_010F	

Address	Register name
0x4002_0110	Reserved
0x4002_0111	
0x4002_0112	
0x4002_0113	
0x4002_0114	Reserved
0x4002_0115	
0x4002_0116	
0x4002_0117	
0x4002_0118	Reserved
0x4002_0119	
0x4002_011A	
0x4002_011B	
0x4002_011C	Reserved
0x4002_011D	
0x4002_011E	
0x4002_011F	

Address	Register name
0x4002_0120	Reserved
0x4002_0121	
0x4002_0122	
0x4002_0123	
0x4002_0124	Reserved
0x4002_0125	
0x4002_0126	
0x4002_0127	
0x4002_0128	Reserved
0x4002_0129	
0x4002_012A	
0x4002_012B	
0x4002_012C	Reserved
0x4002_012D	
0x4002_012E	
0x4002_012F	

Address	Register name
0x4002_0130	Reserved
0x4002_0131	
0x4002_0132	
0x4002_0133	
0x4002_0134	
0x4002_0135	
0x4002_0136	
0x4002_0137	
0x4002_0138	
0x4002_0139	
0x4002_013A	
0x4002_013B	
0x4002_013C	
0x4002_013D	
0x4002_013E	
0x4002_013F	

[5] Serial interface (UART/SIO) [2/2]

<SIO3>

Address	Register name
0x4002_0140	Reserved
0x4002_0141	
0x4002_0142	
0x4002_0143	
0x4002_0144	Reserved
0x4002_0145	
0x4002_0146	
0x4002_0147	
0x4002_0148	Reserved
0x4002_0149	
0x4002_014A	
0x4002_014B	
0x4002_014C	Reserved
0x4002_014D	
0x4002_014E	
0x4002_014F	

Address	Register name
0x4002_0150	Reserved
0x4002_0151	
0x4002_0152	
0x4002_0153	
0x4002_0154	Reserved
0x4002_0155	
0x4002_0156	
0x4002_0157	
0x4002_0158	Reserved
0x4002_0159	
0x4002_015A	
0x4002_015B	
0x4002_015C	Reserved
0x4002_015D	
0x4002_015E	
0x4002_015F	

Address	Register name
0x4002_0160	Reserved
0x4002_0161	
0x4002_0162	
0x4002_0163	
0x4002_0164	Reserved
0x4002_0165	
0x4002_0166	
0x4002_0167	
0x4002_0168	Reserved
0x4002_0169	
0x4002_016A	
0x4002_016B	
0x4002_016C	Reserved
0x4002_016D	
0x4002_016E	
0x4002_016F	

Address	Register name
0x4002_0170	Reserved
0x4002_0171	
0x4002_0172	
0x4002_0173	
0x4002_0174	
0x4002_0175	
0x4002_0176	
0x4002_0177	
0x4002_0178	
0x4002_0179	
0x4002_017A	
0x4002_017B	
0x4002_017C	
0x4002_017D	
0x4002_017E	
0x4002_017F	

<SIO4>

Address	Register name
0x4002_0180	SC4EN
0x4002_0181	<R0>
0x4002_0182	<R0>
0x4002_0183	<R0>
0x4002_0184	SC4BUF
0x4002_0185	<R0>
0x4002_0186	<R0>
0x4002_0187	<R0>
0x4002_0188	SC4CR
0x4002_0189	<R0>
0x4002_018A	<R0>
0x4002_018B	<R0>
0x4002_018C	SC4MOD0
0x4002_018D	<R0>
0x4002_018E	<R0>
0x4002_018F	<R0>

Address	Register name
0x4002_0190	SC4BRCR
0x4002_0191	<R0>
0x4002_0192	<R0>
0x4002_0193	<R0>
0x4002_0194	SC4BRADD
0x4002_0195	<R0>
0x4002_0196	<R0>
0x4002_0197	<R0>
0x4002_0198	SC4MOD1
0x4002_0199	<R0>
0x4002_019A	<R0>
0x4002_019B	<R0>
0x4002_019C	SC4MOD2
0x4002_019D	<R0>
0x4002_019E	<R0>
0x4002_019F	<R0>

Address	Register name
0x4002_01A0	SC4RFC
0x4002_01A1	<R0>
0x4002_01A2	<R0>
0x4002_01A3	<R0>
0x4002_01A4	SC4TFC
0x4002_01A5	<R0>
0x4002_01A6	<R0>
0x4002_01A7	<R0>
0x4002_01A8	SC4RST
0x4002_01A9	<R0>
0x4002_01AA	<R0>
0x4002_01AB	<R0>
0x4002_01AC	SC4TST
0x4002_01AD	<R0>
0x4002_01AE	<R0>
0x4002_01AF	<R0>

Address	Register name
0x4002_01B0	SC4FCNF
0x4002_01B1	<R0>
0x4002_01B2	<R0>
0x4002_01B3	<R0>
0x4002_01B4	
0x4002_01B5	
0x4002_01B6	
0x4002_01B7	
0x4002_01B8	
0x4002_01B9	
0x4002_01BA	
0x4002_01BB	
0x4002_01BC	
0x4002_01BD	
0x4002_01BE	
0x4002_01BF	

25.2.6 [6] 12-bit A/D converter (A/DC) [1/2]

<ADC>

Address	Register name
0x4003_0000	ADCLK
0x4003_0001	
0x4003_0002	
0x4003_0003	
0x4003_0004	ADMOD0
0x4003_0005	
0x4003_0006	
0x4003_0007	
0x4003_0008	ADMOD1
0x4003_0009	
0x4003_000A	
0x4003_000B	
0x4003_000C	ADMOD2
0x4003_000D	
0x4003_000E	
0x4003_000F	

Address	Register name
0x4003_0010	ADCMPCR0
0x4003_0011	
0x4003_0012	
0x4003_0013	
0x4003_0014	ADCMPCR1
0x4003_0015	
0x4003_0016	
0x4003_0017	
0x4003_0018	ADCMP0
0x4003_0019	
0x4003_001A	
0x4003_001B	
0x4003_001C	ADCMP1
0x4003_001D	
0x4003_001E	
0x4003_001F	

Address	Register name
0x4003_0020	AddressEG0
0x4003_0021	
0x4003_0022	
0x4003_0023	
0x4003_0024	AddressEG1
0x4003_0025	
0x4003_0026	
0x4003_0027	
0x4003_0028	AddressEG2
0x4003_0029	
0x4003_002A	
0x4003_002B	
0x4003_002C	AddressEG3
0x4003_002D	
0x4003_002E	
0x4003_002F	

Address	Register name
0x4003_0030	AddressEG4
0x4003_0031	
0x4003_0032	
0x4003_0033	
0x4003_0034	AddressEG5
0x4003_0035	
0x4003_0036	
0x4003_0037	
0x4003_0038	AddressEG6
0x4003_0039	
0x4003_003A	
0x4003_003B	
0x4003_003C	AddressEG7
0x4003_003D	
0x4003_003E	
0x4003_003F	

Address	Register name
0x4003_0040	AddressEG8
0x4003_0041	
0x4003_0042	
0x4003_0043	
0x4003_0044	AddressEG9
0x4003_0045	
0x4003_0046	
0x4003_0047	
0x4003_0048	AddressEG10
0x4003_0049	
0x4003_004A	
0x4003_004B	
0x4003_004C	AddressEG11
0x4003_004D	
0x4003_004E	
0x4003_004F	

Address	Register name
0x4003_0050	ADPSEL0
0x4003_0051	
0x4003_0052	
0x4003_0053	
0x4003_0054	ADPSEL1
0x4003_0055	
0x4003_0056	
0x4003_0057	
0x4003_0058	ADPSEL2
0x4003_0059	
0x4003_005A	
0x4003_005B	
0x4003_005C	ADPSEL3
0x4003_005D	
0x4003_005E	
0x4003_005F	

Address	Register name
0x4003_0060	Reserved
0x4003_0061	
0x4003_0062	
0x4003_0063	
0x4003_0064	Reserved
0x4003_0065	
0x4003_0066	
0x4003_0067	
0x4003_0068	Reserved
0x4003_0069	
0x4003_006A	
0x4003_006B	
0x4003_006C	Reserved
0x4003_006D	
0x4003_006E	
0x4003_006F	

Address	Register name
0x4003_0070	Reserved
0x4003_0071	
0x4003_0072	
0x4003_0073	
0x4003_0074	Reserved
0x4003_0075	
0x4003_0076	
0x4003_0077	
0x4003_0078	Reserved
0x4003_0079	
0x4003_007A	
0x4003_007B	
0x4003_007C	Reserved
0x4003_007D	
0x4003_007E	
0x4003_007F	

Address	Register name
0x4003_0080	ADPINTS0
0x4003_0081	
0x4003_0082	
0x4003_0083	
0x4003_0084	ADPINTS1
0x4003_0085	
0x4003_0086	
0x4003_0087	
0x4003_0088	ADPINTS2
0x4003_0089	
0x4003_008A	
0x4003_008B	
0x4003_008C	ADPINTS3
0x4003_008D	
0x4003_008E	
0x4003_008F	

Address	Register name
0x4003_0090	ADPINTS4
0x4003_0091	
0x4003_0092	
0x4003_0093	
0x4003_0094	ADPINTS5
0x4003_0095	
0x4003_0096	
0x4003_0097	
0x4003_0098	ADPSET0
0x4003_0099	
0x4003_009A	
0x4003_009B	
0x4003_009C	ADPSET1
0x4003_009D	
0x4003_009E	
0x4003_009F	

Address	Register name
0x4003_00A0	ADPSET2
0x4003_00A1	
0x4003_00A2	
0x4003_00A3	
0x4003_00A4	ADPSET3
0x4003_00A5	
0x4003_00A6	
0x4003_00A7	
0x4003_00A8	ADPSET4
0x4003_00A9	
0x4003_00AA	
0x4003_00AB	
0x4003_00AC	ADPSET5
0x4003_00AD	
0x4003_00AE	
0x4003_00AF	

Address	Register name
0x4003_00B0	ADTSET03
0x4003_00B1	
0x4003_00B2	
0x4003_00B3	
0x4003_00B4	ADTSET47
0x4003_00B5	
0x4003_00B6	
0x4003_00B7	
0x4003_00B8	ADTSET811
0x4003_00B9	
0x4003_00BA	
0x4003_00BB	
0x4003_00BC	ADSSET03
0x4003_00BD	
0x4003_00BE	
0x4003_00BF	

[6] 12-bit A/D converter (A/DC) [2/2]

Address	Register name
0x4003_00C0	ADSSET47
0x4003_00C1	
0x4003_00C2	
0x4003_00C3	
0x4003_00C4	ADSSET811
0x4003_00C5	
0x4003_00C6	
0x4003_00C7	
0x4003_00C8	ADASET03
0x4003_00C9	
0x4003_00CA	
0x4003_00CB	
0x4003_00CC	ADASET47
0x4003_00CD	
0x4003_00CE	
0x4003_00CF	

Address	Register name
0x4003_00D0	ADASET811
0x4003_00D1	
0x4003_00D2	
0x4003_00D3	
0x4003_00D4	ADMOD3
0x4003_00D5	
0x4003_00D6	
0x4003_00D7	
0x4003_00D8	
0x4003_00D9	
0x4003_00DA	
0x4003_00DB	
0x4003_00DC	
0x4003_00DD	
0x4003_00DE	
0x4003_00DF	

Address	Register name
0x4003_00E0	
0x4003_00E1	
0x4003_00E2	
0x4003_00E3	
0x4003_00E4	
0x4003_00E5	
0x4003_00E6	
0x4003_00E7	
0x4003_00E8	
0x4003_00E9	
0x4003_00EA	
0x4003_00EB	
0x4003_00EC	
0x4003_00ED	
0x4003_00EE	
0x4003_00EF	

Address	Register name
0x4003_00F0	
0x4003_00F1	
0x4003_00F2	
0x4003_00F3	
0x4003_00F4	
0x4003_00F5	
0x4003_00F6	
0x4003_00F7	
0x4003_00F8	
0x4003_00F9	
0x4003_00FA	
0x4003_00FB	
0x4003_00FC	
0x4003_00FD	
0x4003_00FE	
0x4003_00FF	

25.2.7 [7] Watchdog timer (WDT)

Address	Register name
0x4004_0000	WDMOD
0x4004_0001	<R0>
0x4004_0002	<R0>
0x4004_0003	<R0>
0x4004_0004	WDCR
0x4004_0005	<R0>
0x4004_0006	<R0>
0x4004_0007	<R0>
0x4004_0008	
0x4004_0009	
0x4004_000A	
0x4004_000B	
0x4004_000C	
0x4004_000D	
0x4004_000E	
0x4004_000F	

Address	Register name
0x4004_0010	
0x4004_0011	
0x4004_0012	
0x4004_0013	
0x4004_0014	
0x4004_0015	
0x4004_0016	
0x4004_0017	
0x4004_0018	
0x4004_0019	
0x4004_001A	
0x4004_001B	
0x4004_001C	
0x4004_001D	
0x4004_001E	
0x4004_001F	

Address	Register name
0x4004_0020	
0x4004_0021	
0x4004_0022	
0x4004_0023	
0x4004_0024	
0x4004_0025	
0x4004_0026	
0x4004_0027	
0x4004_0028	
0x4004_0029	
0x4004_002A	
0x4004_002B	
0x4004_002C	
0x4004_002D	
0x4004_002E	
0x4004_002F	

Address	Register name
0x4004_0030	
0x4004_0031	
0x4004_0032	
0x4004_0033	
0x4004_0034	
0x4004_0035	
0x4004_0036	
0x4004_0037	
0x4004_0038	
0x4004_0039	
0x4004_003A	
0x4004_003B	
0x4004_003C	
0x4004_003D	
0x4004_003E	
0x4004_003F	

25.2.8 [8] Real time clock (RTC)

Address	Register name
0x4004_0100	SECR
0x4004_0101	MINR
0x4004_0102	HOURR
0x4004_0103	
0x4004_0104	DAYR
0x4004_0105	DATER
0x4004_0106	MONTHR
0x4004_0107	YEARR
0x4004_0108	PAGER
0x4004_0109	
0x4004_010A	
0x4004_010B	
0x4004_010C	RESTR
0x4004_010D	
0x4004_010E	
0x4004_010F	

Address	Register name
0x4004_0110	
0x4004_0111	
0x4004_0112	
0x4004_0113	
0x4004_0114	
0x4004_0115	
0x4004_0116	
0x4004_0117	
0x4004_0118	
0x4004_0119	
0x4004_011A	
0x4004_011B	
0x4004_011C	
0x4004_011D	
0x4004_011E	
0x4004_011F	

Address	Register name
0x4004_0120	
0x4004_0121	
0x4004_0122	
0x4004_0123	
0x4004_0124	
0x4004_0125	
0x4004_0126	
0x4004_0127	
0x4004_0128	
0x4004_0129	
0x4004_012A	
0x4004_012B	
0x4004_012C	
0x4004_012D	
0x4004_012E	
0x4004_012F	

Address	Register name
0x4004_0130	
0x4004_0131	
0x4004_0132	
0x4004_0133	
0x4004_0134	
0x4004_0135	
0x4004_0136	
0x4004_0137	
0x4004_0138	
0x4004_0139	
0x4004_013A	
0x4004_013B	
0x4004_013C	
0x4004_013D	
0x4004_013E	
0x4004_013F	

25.2.9 [9] Clock generator (CG)

Address	Register name
0x4004_0200	CGSYSCR
0x4004_0201	
0x4004_0202	
0x4004_0203	<R0>
0x4004_0204	CGOSCCR
0x4004_0205	
0x4004_0206	
0x4004_0207	
0x4004_0208	CGSTBYCR
0x4004_0209	
0x4004_020A	
0x4004_020B	<R0>
0x4004_020C	CGPLLSEL
0x4004_020D	
0x4004_020E	<R0>
0x4004_020F	<R0>

Address	Register name
0x4004_0210	CGCKSEL
0x4004_0211	<R0>
0x4004_0212	<R0>
0x4004_0213	<R0>
0x4004_0214	CGICRCG
0x4004_0215	<R0>
0x4004_0216	<R0>
0x4004_0217	<R0>
0x4004_0218	CGNMIFLG
0x4004_0219	<R0>
0x4004_021A	<R0>
0x4004_021B	<R0>
0x4004_021C	CGRSTFLG
0x4004_021D	<R0>
0x4004_021E	<R0>
0x4004_021F	<R0>

Address	Register name
0x4004_0220	CGIMCGA
0x4004_0221	
0x4004_0222	
0x4004_0223	
0x4004_0224	CGIMCGB
0x4004_0225	
0x4004_0226	
0x4004_0227	
0x4004_0228	CGIMCGC
0x4004_0229	
0x4004_022A	
0x4004_022B	
0x4004_022C	CGIMCGD
0x4004_022D	
0x4004_022E	
0x4004_022F	

Address	Register name
0x4004_0230	CGIMCGE
0x4004_0231	
0x4004_0232	
0x4004_0233	
0x4004_0234	
0x4004_0235	
0x4004_0236	
0x4004_0237	
0x4004_0238	
0x4004_0239	
0x4004_023A	
0x4004_023B	
0x4004_023C	
0x4004_023D	
0x4004_023E	
0x4004_023F	

25.2.10 [10] Remote control signal preprocessor (RMC)

Address	Register name
0x4004_0400	RMCEN
0x4004_0401	<R0>
0x4004_0402	<R0>
0x4004_0403	<R0>
0x4004_0404	RMCREN
0x4004_0405	<R0>
0x4004_0406	<R0>
0x4004_0407	<R0>
0x4004_0408	RMCRBUF1
0x4004_0409	
0x4004_040A	
0x4004_040B	
0x4004_040C	RMCRBUF2
0x4004_040D	
0x4004_040E	
0x4004_040F	

Address	Register name
0x4004_0410	RMCRBUF3
0x4004_0411	
0x4004_0412	
0x4004_0413	
0x4004_0414	RMCRCR1
0x4004_0415	
0x4004_0416	
0x4004_0417	
0x4004_0418	RMCRCR2
0x4004_0419	
0x4004_041A	
0x4004_041B	
0x4004_041C	RMCRCR3
0x4004_041D	
0x4004_041E	<R0>
0x4004_041F	<R0>

Address	Register name
0x4004_0420	RMCCR4
0x4004_0421	<R0>
0x4004_0422	<R0>
0x4004_0423	<R0>
0x4004_0424	RMCRSTAT
0x4004_0425	
0x4004_0426	<R0>
0x4004_0427	<R0>
0x4004_0428	RMCEND1
0x4004_0429	
0x4004_042A	
0x4004_042B	
0x4004_042C	RMCEND2
0x4004_042D	
0x4004_042E	
0x4004_042F	

Address	Register name
0x4004_0430	RMCEND3
0x4004_0431	
0x4004_0432	
0x4004_0433	
0x4004_0434	RMCFSSEL
0x4004_0435	
0x4004_0436	
0x4004_0437	
0x4004_0438	
0x4004_0439	
0x4004_043A	
0x4004_043B	
0x4004_043C	
0x4004_043D	
0x4004_043E	
0x4004_043F	

25.2.11 [11] Oscillation frequency detector (OFD)

Address	Register name
0x4004_0800	OFDCR1
0x4004_0801	
0x4004_0802	
0x4004_0803	
0x4004_0804	OFDCR2
0x4004_0805	
0x4004_0806	
0x4004_0807	
0x4004_0808	OFDMN
0x4004_0809	
0x4004_080A	
0x4004_080B	
0x4004_080C	
0x4004_080D	
0x4004_080E	
0x4004_080F	

Address	Register name
0x4004_0810	OFDMX
0x4004_0811	
0x4004_0812	
0x4004_0813	
0x4004_0814	
0x4004_0815	
0x4004_0816	
0x4004_0817	
0x4004_0818	OFDRST
0x4004_0819	
0x4004_081A	
0x4004_081B	
0x4004_081C	OFDSTAT
0x4004_081D	
0x4004_081E	
0x4004_081F	

Address	Register name
0x4004_0820	
0x4004_0821	
0x4004_0822	
0x4004_0823	
0x4004_0824	
0x4004_0825	
0x4004_0826	
0x4004_0827	
0x4004_0828	
0x4004_0829	
0x4004_082A	
0x4004_082B	
0x4004_082C	
0x4004_082D	
0x4004_082E	
0x4004_082F	

Address	Register name
0x4004_0830	
0x4004_0831	
0x4004_0832	
0x4004_0833	
0x4004_0834	
0x4004_0835	
0x4004_0836	
0x4004_0837	
0x4004_0838	
0x4004_0839	
0x4004_083A	
0x4004_083B	
0x4004_083C	
0x4004_083D	
0x4004_083E	
0x4004_083F	

25.2.12 [12] Power on reset (POR), Voltage detecting circuit (VLTD)

Address	Register name
0x4004_0900	VDCR
0x4004_0901	
0x4004_0902	
0x4004_0903	
0x4004_0904	VDSR
0x4004_0905	
0x4004_0906	
0x4004_0907	
0x4004_0908	
0x4004_0909	
0x4004_090A	
0x4004_090B	
0x4004_090C	
0x4004_090D	
0x4004_090E	
0x4004_090F	

Address	Register name
0x4004_0910	
0x4004_0911	
0x4004_0912	
0x4004_0913	
0x4004_0914	
0x4004_0915	
0x4004_0916	
0x4004_0917	
0x4004_0918	
0x4004_0919	
0x4004_091A	
0x4004_091B	
0x4004_091C	
0x4004_091D	
0x4004_091E	
0x4004_091F	

Address	Register name
0x4004_0920	
0x4004_0921	
0x4004_0922	
0x4004_0923	
0x4004_0924	
0x4004_0925	
0x4004_0926	
0x4004_0927	
0x4004_0928	
0x4004_0929	
0x4004_092A	
0x4004_092B	
0x4004_092C	
0x4004_092D	
0x4004_092E	
0x4004_092F	

Address	Register name
0x4004_0930	
0x4004_0931	
0x4004_0932	
0x4004_0933	
0x4004_0934	
0x4004_0935	
0x4004_0936	
0x4004_0937	
0x4004_0938	
0x4004_0939	
0x4004_093A	
0x4004_093B	
0x4004_093C	
0x4004_093D	
0x4004_093E	
0x4004_093F	

25.2.13 [13] Multi purpose timer (PMD)[1/2]

Address	Register name
0x4005_0400	MTPD0MDEN
0x4005_0401	<R0>
0x4005_0402	<R0>
0x4005_0403	<R0>
0x4005_0404	MTPD0PORTMD
0x4005_0405	<R0>
0x4005_0406	<R0>
0x4005_0407	<R0>
0x4005_0408	MTPD0MDCR
0x4005_0409	<R0>
0x4005_040A	<R0>
0x4005_040B	<R0>
0x4005_040C	MTPD0CNTSTA
0x4005_040D	<R0>
0x4005_040E	<R0>
0x4005_040F	<R0>

Address	Register name
0x4005_0410	MTPD0MDCNT
0x4005_0411	
0x4005_0412	<R0>
0x4005_0413	<R0>
0x4005_0414	MTPD0MDPRD
0x4005_0415	
0x4005_0416	<R0>
0x4005_0417	<R0>
0x4005_0418	MTPD0CMPU
0x4005_0419	
0x4005_041A	<R0>
0x4005_041B	<R0>
0x4005_041C	MTPD0CMPV
0x4005_041D	
0x4005_041E	<R0>
0x4005_041F	<R0>

Address	Register name
0x4005_0420	MTPD0CMPW
0x4005_0421	
0x4005_0422	<R0>
0x4005_0423	<R0>
0x4005_0424	Reserved
0x4005_0425	Reserved
0x4005_0426	Reserved
0x4005_0427	Reserved
0x4005_0428	MTPD0MDOUT
0x4005_0429	
0x4005_042A	<R0>
0x4005_042B	<R0>
0x4005_042C	MTPD0MDPOT
0x4005_042D	<R0>
0x4005_042E	<R0>
0x4005_042F	<R0>

Address	Register name
0x4005_0430	MTPD0EMGREL
0x4005_0431	<R0>
0x4005_0432	<R0>
0x4005_0433	<R0>
0x4005_0434	MTPD0EMGCR
0x4005_0435	
0x4005_0436	<R0>
0x4005_0437	<R0>
0x4005_0438	MTPD0EMGST
0x4005_0439	<R0>
0x4005_043A	<R0>
0x4005_043B	<R0>
0x4005_043C	Reserved
0x4005_043D	Reserved
0x4005_043E	Reserved
0x4005_043F	Reserved

Address	Register name
0x4005_0440	Reserved
0x4005_0441	Reserved
0x4005_0442	Reserved
0x4005_0443	Reserved
0x4005_0444	MTPD0DTR
0x4005_0445	<R0>
0x4005_0446	<R0>
0x4005_0447	<R0>
0x4005_0448	MTPD0TRGCMP0
0x4005_0449	
0x4005_044A	<R0>
0x4005_044B	<R0>
0x4005_044C	MTPD0TRGCMP1
0x4005_044D	
0x4005_044E	<R0>
0x4005_044F	<R0>

Address	Register name
0x4005_0450	Reserved
0x4005_0451	Reserved
0x4005_0452	Reserved
0x4005_0453	Reserved
0x4005_0454	Reserved
0x4005_0455	Reserved
0x4005_0456	Reserved
0x4005_0457	Reserved
0x4005_0458	MTPD0TRGCR
0x4005_0459	
0x4005_045A	<R0>
0x4005_045B	<R0>
0x4005_045C	MTPD0TRGMD
0x4005_045D	<R0>
0x4005_045E	<R0>
0x4005_045F	<R0>

Address	Register name
0x4005_0460	Reserved
0x4005_0461	Reserved
0x4005_0462	Reserved
0x4005_0463	Reserved
0x4005_0464	Reserved
0x4005_0465	
0x4005_0466	
0x4005_0467	
0x4005_0468	Reserved
0x4005_0469	
0x4005_046A	
0x4005_046B	
0x4005_046C	Reserved
0x4005_046D	
0x4005_046E	
0x4005_046F	

Address	Register name
0x4005_0470	Reserved
0x4005_0471	
0x4005_0472	
0x4005_0473	
0x4005_0474	Reserved
0x4005_0475	
0x4005_0476	
0x4005_0477	
0x4005_0478	Reserved
0x4005_0479	
0x4005_047A	
0x4005_047B	
0x4005_047C	Reserved
0x4005_047D	
0x4005_047E	
0x4005_047F	

[13] Multi purpose timer (PMD) [2/2]

Address	Register name
0x4005_0480	Reserved
0x4005_0481	
0x4005_0482	
0x4005_0483	
0x4005_0484	Reserved
0x4005_0485	
0x4005_0486	
0x4005_0487	
0x4005_0488	Reserved
0x4005_0489	
0x4005_048A	
0x4005_048B	
0x4005_048C	Reserved
0x4005_048D	
0x4005_048E	
0x4005_048F	

Address	Register name
0x4005_0490	Reserved
0x4005_0491	
0x4005_0492	
0x4005_0493	
0x4005_0494	Reserved
0x4005_0495	
0x4005_0496	
0x4005_0497	
0x4005_0498	Reserved
0x4005_0499	
0x4005_049A	
0x4005_049B	
0x4005_049C	Reserved
0x4005_049D	
0x4005_049E	
0x4005_049F	

Address	Register name
0x4005_04A0	Reserved
0x4005_04A1	
0x4005_04A2	
0x4005_04A3	
0x4005_04A4	Reserved
0x4005_04A5	
0x4005_04A6	
0x4005_04A7	
0x4005_04A8	Reserved
0x4005_04A9	
0x4005_04AA	
0x4005_04AB	
0x4005_04AC	Reserved
0x4005_04AD	
0x4005_04AE	
0x4005_04AF	

Address	Register name
0x4005_04B0	Reserved
0x4005_04B1	
0x4005_04B2	
0x4005_04B3	
0x4005_04B4	Reserved
0x4005_04B5	
0x4005_04B6	
0x4005_04B7	
0x4005_04B8	Reserved
0x4005_04B9	
0x4005_04BA	
0x4005_04BB	
0x4005_04BC	Reserved
0x4005_04BD	
0x4005_04BE	
0x4005_04BF	

Address	Register name
0x4005_04C0	Reserved
0x4005_04C1	
0x4005_04C2	
0x4005_04C3	
0x4005_04C4	Reserved
0x4005_04C5	
0x4005_04C6	
0x4005_04C7	
0x4005_04C8	Reserved
0x4005_04C9	
0x4005_04CA	
0x4005_04CB	
0x4005_04CC	Reserved
0x4005_04CD	
0x4005_04CE	
0x4005_04CF	

Address	Register name
0x4005_04D0	Reserved
0x4005_04D1	
0x4005_04D2	
0x4005_04D3	
0x4005_04D4	Reserved
0x4005_04D5	
0x4005_04D6	
0x4005_04D7	
0x4005_04D8	Reserved
0x4005_04D9	
0x4005_04DA	
0x4005_04DB	
0x4005_04DC	Reserved
0x4005_04DD	
0x4005_04DE	
0x4005_04DF	

Address	Register name
0x4005_04E0	Reserved
0x4005_04E1	
0x4005_04E2	
0x4005_04E3	
0x4005_04E4	Reserved
0x4005_04E5	
0x4005_04E6	
0x4005_04E7	
0x4005_04E8	Reserved
0x4005_04E9	
0x4005_04EA	
0x4005_04EB	
0x4005_04EC	Reserved
0x4005_04ED	
0x4005_04EE	
0x4005_04EF	

Address	Register name
0x4005_04F0	Reserved
0x4005_04F1	
0x4005_04F2	
0x4005_04F3	
0x4005_04F4	Reserved
0x4005_04F5	
0x4005_04F6	
0x4005_04F7	
0x4005_04F8	Reserved
0x4005_04F9	
0x4005_04FA	
0x4005_04FB	
0x4005_04FC	Reserved
0x4005_04FD	
0x4005_04FE	
0x4005_04FF	

[13] Multi purpose timer (TMR/IGBT) [1/3]

<MPT0>

Address	Register name
0x4005_0800	MT0EN
0x4005_0801	
0x4005_0802	
0x4005_0803	
0x4005_0804	MT0RUN
0x4005_0805	
0x4005_0806	
0x4005_0807	
0x4005_0808	MT0TBCR
0x4005_0809	
0x4005_080A	
0x4005_080B	
0x4005_080C	MT0TBMOD
0x4005_080D	
0x4005_080E	
0x4005_080F	

Address	Register name
0x4005_0810	MT0TBFFCR
0x4005_0811	
0x4005_0812	
0x4005_0813	
0x4005_0814	MT0TBST
0x4005_0815	
0x4005_0816	
0x4005_0817	
0x4005_0818	MT0TBIM
0x4005_0819	
0x4005_081A	
0x4005_081B	
0x4005_081C	MT0TBUC
0x4005_081D	
0x4005_081E	
0x4005_081F	

Address	Register name
0x4005_0820	MT0RG0
0x4005_0821	
0x4005_0822	
0x4005_0823	
0x4005_0824	MT0RG1
0x4005_0825	
0x4005_0826	
0x4005_0827	
0x4005_0828	MT0CP0
0x4005_0829	
0x4005_082A	
0x4005_082B	
0x4005_082C	MT0CP1
0x4005_082D	
0x4005_082E	
0x4005_082F	

Address	Register name
0x4005_0830	MT0IGCR
0x4005_0831	
0x4005_0832	
0x4005_0833	
0x4005_0834	MT0IGRESTA
0x4005_0835	
0x4005_0836	
0x4005_0837	
0x4005_0838	MT0IGST
0x4005_0839	
0x4005_083A	
0x4005_083B	
0x4005_083C	MT0IGICR
0x4005_083D	
0x4005_083E	
0x4005_083F	

Address	Register name
0x4005_0840	MT0IGOCR
0x4005_0841	
0x4005_0842	
0x4005_0843	
0x4005_0844	MT0IGRG2
0x4005_0845	
0x4005_0846	
0x4005_0847	
0x4005_0848	MT0IGRG3
0x4005_0849	
0x4005_084A	
0x4005_084B	
0x4005_084C	MT0IGRG4
0x4005_084D	
0x4005_084E	
0x4005_084F	

Address	Register name
0x4005_0850	MT0IGEMGCR
0x4005_0851	
0x4005_0852	
0x4005_0853	
0x4005_0854	MT0IGEMGST
0x4005_0855	
0x4005_0856	
0x4005_0857	
0x4005_0858	Reserved
0x4005_0859	
0x4005_085A	
0x4005_085B	
0x4005_085C	Reserved
0x4005_085D	
0x4005_085E	
0x4005_085F	

Address	Register name
0x4005_0860	Reserved
0x4005_0861	
0x4005_0862	
0x4005_0863	
0x4005_0864	Reserved
0x4005_0865	
0x4005_0866	
0x4005_0867	
0x4005_0868	Reserved
0x4005_0869	
0x4005_086A	
0x4005_086B	
0x4005_086C	Reserved
0x4005_086D	
0x4005_086E	
0x4005_086F	

Address	Register name
0x4005_0870	Reserved
0x4005_0871	
0x4005_0872	
0x4005_0873	
0x4005_0874	Reserved
0x4005_0875	
0x4005_0876	
0x4005_0877	
0x4005_0878	Reserved
0x4005_0879	
0x4005_087A	
0x4005_087B	
0x4005_087C	Reserved
0x4005_087D	
0x4005_087E	
0x4005_087F	

[13] Multi purpose timer (TMR/IGBT) [2/3]

Address	Register name
0x4005_0880	Reserved
0x4005_0881	
0x4005_0882	
0x4005_0883	
0x4005_0884	Reserved
0x4005_0885	
0x4005_0886	
0x4005_0887	
0x4005_0888	Reserved
0x4005_0889	
0x4005_088A	
0x4005_088B	
0x4005_088C	Reserved
0x4005_088D	
0x4005_088E	
0x4005_088F	

Address	Register name
0x4005_0890	Reserved
0x4005_0891	
0x4005_0892	
0x4005_0893	
0x4005_0894	Reserved
0x4005_0895	
0x4005_0896	
0x4005_0897	
0x4005_0898	Reserved
0x4005_0899	
0x4005_089A	
0x4005_089B	
0x4005_089C	Reserved
0x4005_089D	
0x4005_089E	
0x4005_089F	

Address	Register name
0x4005_08A0	Reserved
0x4005_08A1	
0x4005_08A2	
0x4005_08A3	
0x4005_08A4	Reserved
0x4005_08A5	
0x4005_08A6	
0x4005_08A7	
0x4005_08A8	Reserved
0x4005_08A9	
0x4005_08AA	
0x4005_08AB	
0x4005_08AC	Reserved
0x4005_08AD	
0x4005_08AE	
0x4005_08AF	

Address	Register name
0x4005_08B0	Reserved
0x4005_08B1	
0x4005_08B2	
0x4005_08B3	
0x4005_08B4	Reserved
0x4005_08B5	
0x4005_08B6	
0x4005_08B7	
0x4005_08B8	Reserved
0x4005_08B9	
0x4005_08BA	
0x4005_08BB	
0x4005_08BC	Reserved
0x4005_08BD	
0x4005_08BE	
0x4005_08BF	

Address	Register name
0x4005_08C0	Reserved
0x4005_08C1	
0x4005_08C2	
0x4005_08C3	
0x4005_08C4	Reserved
0x4005_08C5	
0x4005_08C6	
0x4005_08C7	
0x4005_08C8	Reserved
0x4005_08C9	
0x4005_08CA	
0x4005_08CB	
0x4005_08CC	Reserved
0x4005_08CD	
0x4005_08CE	
0x4005_08CF	

Address	Register name
0x4005_08D0	Reserved
0x4005_08D1	
0x4005_08D2	
0x4005_08D3	
0x4005_08D4	Reserved
0x4005_08D5	
0x4005_08D6	
0x4005_08D7	
0x4005_08D8	Reserved
0x4005_08D9	
0x4005_08DA	
0x4005_08DB	
0x4005_08DC	Reserved
0x4005_08DD	
0x4005_08DE	
0x4005_08DF	

Address	Register name
0x4005_08E0	Reserved
0x4005_08E1	
0x4005_08E2	
0x4005_08E3	
0x4005_08E4	Reserved
0x4005_08E5	
0x4005_08E6	
0x4005_08E7	
0x4005_08E8	Reserved
0x4005_08E9	
0x4005_08EA	
0x4005_08EB	
0x4005_08EC	Reserved
0x4005_08ED	
0x4005_08EE	
0x4005_08EF	

Address	Register name
0x4005_08F0	Reserved
0x4005_08F1	
0x4005_08F2	
0x4005_08F3	
0x4005_08F4	Reserved
0x4005_08F5	
0x4005_08F6	
0x4005_08F7	
0x4005_08F8	Reserved
0x4005_08F9	
0x4005_08FA	
0x4005_08FB	
0x4005_08FC	Reserved
0x4005_08FD	
0x4005_08FE	
0x4005_08FF	

[13] Multi purpose timer (TMR/IGBT) [3/3]

Address	Register name
0x4005_0900	Reserved
0x4005_0901	
0x4005_0902	
0x4005_0903	
0x4005_0904	Reserved
0x4005_0905	
0x4005_0906	
0x4005_0907	
0x4005_0908	Reserved
0x4005_0909	
0x4005_090A	
0x4005_090B	
0x4005_090C	Reserved
0x4005_090D	
0x4005_090E	
0x4005_090F	

Address	Register name
0x4005_0910	Reserved
0x4005_0911	
0x4005_0912	
0x4005_0913	
0x4005_0914	Reserved
0x4005_0915	
0x4005_0916	
0x4005_0917	
0x4005_0918	Reserved
0x4005_0919	
0x4005_091A	
0x4005_091B	
0x4005_091C	Reserved
0x4005_091D	
0x4005_091E	
0x4005_091F	

Address	Register name
0x4005_0920	Reserved
0x4005_0921	
0x4005_0922	
0x4005_0923	
0x4005_0924	Reserved
0x4005_0925	
0x4005_0926	
0x4005_0927	
0x4005_0928	Reserved
0x4005_0929	
0x4005_092A	
0x4005_092B	
0x4005_092C	Reserved
0x4005_092D	
0x4005_092E	
0x4005_092F	

Address	Register name
0x4005_0930	Reserved
0x4005_0931	
0x4005_0932	
0x4005_0933	
0x4005_0934	Reserved
0x4005_0935	
0x4005_0936	
0x4005_0937	
0x4005_0938	Reserved
0x4005_0939	
0x4005_093A	
0x4005_093B	
0x4005_093C	Reserved
0x4005_093D	
0x4005_093E	
0x4005_093F	

Address	Register name
0x4005_0940	Reserved
0x4005_0941	
0x4005_0942	
0x4005_0943	
0x4005_0944	Reserved
0x4005_0945	
0x4005_0946	
0x4005_0947	
0x4005_0948	Reserved
0x4005_0949	
0x4005_094A	
0x4005_094B	
0x4005_094C	Reserved
0x4005_094D	
0x4005_094E	
0x4005_094F	

Address	Register name
0x4005_0950	Reserved
0x4005_0951	
0x4005_0952	
0x4005_0953	
0x4005_0954	Reserved
0x4005_0955	
0x4005_0956	
0x4005_0957	
0x4005_0958	Reserved
0x4005_0959	
0x4005_095A	
0x4005_095B	
0x4005_095C	Reserved
0x4005_095D	
0x4005_095E	
0x4005_095F	

Address	Register name
0x4005_0960	Reserved
0x4005_0961	
0x4005_0962	
0x4005_0963	
0x4005_0964	Reserved
0x4005_0965	
0x4005_0966	
0x4005_0967	
0x4005_0968	Reserved
0x4005_0969	
0x4005_096A	
0x4005_096B	
0x4005_096C	Reserved
0x4005_096D	
0x4005_096E	
0x4005_096F	

Address	Register name
0x4005_0970	Reserved
0x4005_0971	
0x4005_0972	
0x4005_0973	
0x4005_0974	Reserved
0x4005_0975	
0x4005_0976	
0x4005_0977	
0x4005_0978	Reserved
0x4005_0979	
0x4005_097A	
0x4005_097B	
0x4005_097C	Reserved
0x4005_097D	
0x4005_097E	
0x4005_097F	

25.2.14 [15] DMA controller (DMAC)

<DMAC>

Address	Register name
0x4008_0000	DMACInt
0x4008_0001	Status
0x4008_0002	
0x4008_0003	
0x4008_0004	DMACIntTC
0x4008_0005	Status
0x4008_0006	
0x4008_0007	
0x4008_0008	DMACIntTC
0x4008_0009	Clear
0x4008_000A	
0x4008_000B	
0x4008_000C	DMACInt
0x4008_000D	ErrorStatus
0x4008_000E	
0x4008_000F	

Address	Register name
0x4008_0010	DMACInt
0x4008_0011	ErrClr
0x4008_0012	
0x4008_0013	
0x4008_0014	DMACRawInt
0x4008_0015	TCStatus
0x4008_0016	
0x4008_0017	
0x4008_0018	DMACRawInt
0x4008_0019	ErrorStatus
0x4008_001A	
0x4008_001B	
0x4008_001C	DMACEnbld
0x4008_001D	Chns
0x4008_001E	
0x4008_001F	

Address	Register name
0x4008_0020	DMACSoftB
0x4008_0021	Req
0x4008_0022	
0x4008_0023	
0x4008_0024	DMACSoftS
0x4008_0025	Req
0x4008_0026	
0x4008_0027	
0x4008_0028	Reserved
0x4008_0029	
0x4008_002A	
0x4008_002B	
0x4008_002C	Reserved
0x4008_002D	
0x4008_002E	
0x4008_002F	

Address	Register name
0x4008_0030	DMAC
0x4008_0031	Configuration
0x4008_0032	
0x4008_0033	
0x4008_0034	Reserved
0x4008_0035	
0x4008_0036	
0x4008_0037	
0x4008_0038	Reserved
0x4008_0039	
0x4008_003A	
0x4008_003B	
0x4008_003C	Reserved
0x4008_003D	
0x4008_003E	
0x4008_003F	

Address	Register name
0x4008_0100	DMACC0Src
0x4008_0101	Addr
0x4008_0102	
0x4008_0103	
0x4008_0104	DMACC0Dest
0x4008_0105	Addr
0x4008_0106	
0x4008_0107	
0x4008_0108	DMACC0LLI
0x4008_0109	
0x4008_010A	
0x4008_010B	
0x4008_010C	DMACC0
0x4008_010D	Control
0x4008_010E	
0x4008_010F	

Address	Register name
0x4008_0110	DMACC0
0x4008_0111	Configuration
0x4008_0112	
0x4008_0113	
0x4008_0114	Reserved
0x4008_0115	
0x4008_0116	
0x4008_0117	
0x4008_0118	Reserved
0x4008_0119	
0x4008_011A	
0x4008_011B	
0x4008_011C	Reserved
0x4008_011D	
0x4008_011E	
0x4008_011F	

Address	Register name
0x4008_0120	DMACC1Src
0x4008_0121	Addr
0x4008_0122	
0x4008_0123	
0x4008_0124	DMACC1Dest
0x4008_0125	Addr
0x4008_0126	
0x4008_0127	
0x4008_0128	DMACC1LLI
0x4008_0129	
0x4008_012A	
0x4008_012B	
0x4008_012C	DMACC1
0x4008_012D	Control
0x4008_012E	
0x4008_012F	

Address	Register name
0x4008_0130	DMACC1
0x4008_0131	Configuration
0x4008_0132	
0x4008_0133	
0x4008_0134	Reserved
0x4008_0135	
0x4008_0136	
0x4008_0137	
0x4008_0138	Reserved
0x4008_0139	
0x4008_013A	
0x4008_013B	
0x4008_013C	Reserved
0x4008_013D	
0x4008_013E	
0x4008_013F	

25.2.15 [15] SSP controller

<SSP0>

Address	Register name
0x400C_0000	SSP0CR0
0x400C_0001	
0x400C_0002	
0x400C_0003	
0x400C_0004	SSP0CR1
0x400C_0005	
0x400C_0006	
0x400C_0007	
0x400C_0008	SSP0DR
0x400C_0009	
0x400C_000A	
0x400C_000B	
0x400C_000C	SSP0SR
0x400C_000D	
0x400C_000E	
0x400C_000F	

Address	Register name
0x400C_0010	SSP0CPSR
0x400C_0011	
0x400C_0012	
0x400C_0013	
0x400C_0014	SSP0IMSC
0x400C_0015	
0x400C_0016	
0x400C_0017	
0x400C_0018	SSP0RIS
0x400C_0019	
0x400C_001A	
0x400C_001B	
0x400C_001C	SSP0MIS
0x400C_001D	
0x400C_001E	
0x400C_001F	

Address	Register name
0x400C_0020	SSP0ICR
0x400C_0021	
0x400C_0022	
0x400C_0023	
0x400C_0024	SSP0DMACR
0x400C_0025	
0x400C_0026	
0x400C_0027	
0x400C_0028	Reserved
0x400C_0029	
0x400C_002A	
0x400C_002B	
0x400C_002C	Reserved
0x400C_002D	
0x400C_002E	
0x400C_002F	

Address	Register name
0x400C_0030	Reserved
0x400C_0031	
0x400C_0032	
0x400C_0033	
0x400C_0034	Reserved
0x400C_0035	
0x400C_0036	
0x400C_0037	
0x400C_0038	Reserved
0x400C_0039	
0x400C_003A	
0x400C_003B	
0x400C_003C	Reserved
0x400C_003D	
0x400C_003E	
0x400C_003F	

Address	Register name
0x400C_1000	Reserved
0x400C_1001	
0x400C_1002	
0x400C_1003	
0x400C_1004	Reserved
0x400C_1005	
0x400C_1006	
0x400C_1007	
0x400C_1008	Reserved
0x400C_1009	
0x400C_100A	
0x400C_100B	
0x400C_100C	Reserved
0x400C_100D	
0x400C_100E	
0x400C_100F	

Address	Register name
0x400C_1010	Reserved
0x400C_1011	
0x400C_1012	
0x400C_1013	
0x400C_1014	Reserved
0x400C_1015	
0x400C_1016	
0x400C_1017	
0x400C_1018	Reserved
0x400C_1019	
0x400C_101A	
0x400C_101B	
0x400C_101C	Reserved
0x400C_101D	
0x400C_101E	
0x400C_101F	

Address	Register name
0x400C_1020	Reserved
0x400C_1021	
0x400C_1022	
0x400C_1023	
0x400C_1024	Reserved
0x400C_1025	
0x400C_1026	
0x400C_1027	
0x400C_1028	Reserved
0x400C_1029	
0x400C_102A	
0x400C_102B	
0x400C_102C	Reserved
0x400C_102D	
0x400C_102E	
0x400C_102F	

Address	Register name
0x400C_1030	Reserved
0x400C_1031	
0x400C_1032	
0x400C_1033	
0x400C_1034	Reserved
0x400C_1035	
0x400C_1036	
0x400C_1037	
0x400C_1038	Reserved
0x400C_1039	
0x400C_103A	
0x400C_103B	
0x400C_103C	Reserved
0x400C_103D	
0x400C_103E	
0x400C_103F	

25.2.16 [16] Flash controller

Address	Register name
0x41FF_F000	Reserved
0x41FF_F001	
0x41FF_F002	
0x41FF_F003	
0x41FF_F004	Reserved
0x41FF_F005	
0x41FF_F006	
0x41FF_F007	
0x41FF_F008	Reserved
0x41FF_F009	
0x41FF_F00A	
0x41FF_F00B	
0x41FF_F00C	Reserved
0x41FF_F00D	
0x41FF_F00E	
0x41FF_F00F	

Address	Register name
0x41FF_F010	FCSECBIT
0x41FF_F011	
0x41FF_F012	
0x41FF_F013	
0x41FF_F014	Reserved
0x41FF_F015	
0x41FF_F016	
0x41FF_F017	
0x41FF_F018	Reserved
0x41FF_F019	
0x41FF_F01A	
0x41FF_F01B	
0x41FF_F01C	Reserved
0x41FF_F01D	
0x41FF_F01E	
0x41FF_F01F	

Address	Register name
0x41FF_F020	FCFLCS
0x41FF_F021	
0x41FF_F022	
0x41FF_F023	
0x41FF_F024	Reserved
0x41FF_F025	
0x41FF_F026	
0x41FF_F027	
0x41FF_F028	Reserved
0x41FF_F029	
0x41FF_F02A	
0x41FF_F02B	
0x41FF_F02C	Reserved
0x41FF_F02D	
0x41FF_F02E	
0x41FF_F02F	

Address	Register name
0x41FF_F030	Reserved
0x41FF_F031	
0x41FF_F032	
0x41FF_F033	
0x41FF_F034	Reserved
0x41FF_F035	
0x41FF_F036	
0x41FF_F037	
0x41FF_F038	Reserved
0x41FF_F039	
0x41FF_F03A	
0x41FF_F03B	
0x41FF_F03C	Reserved
0x41FF_F03D	
0x41FF_F03E	
0x41FF_F03F	

Address	Register name
0x41FF_F040	Reserved
0x41FF_F041	
0x41FF_F042	
0x41FF_F043	
0x41FF_F044	Reserved
0x41FF_F045	
0x41FF_F046	
0x41FF_F047	
0x41FF_F048	Reserved
0x41FF_F049	
0x41FF_F04A	
0x41FF_F04B	
0x41FF_F04C	Reserved
0x41FF_F04D	
0x41FF_F04E	
0x41FF_F04F	

Address	Register name
0x41FF_F050	Reserved
0x41FF_F051	
0x41FF_F052	
0x41FF_F053	
0x41FF_F054	Reserved
0x41FF_F055	
0x41FF_F056	
0x41FF_F057	
0x41FF_F058	Reserved
0x41FF_F059	
0x41FF_F05A	
0x41FF_F05B	
0x41FF_F05C	Reserved
0x41FF_F05D	
0x41FF_F05E	
0x41FF_F05F	

Address	Register name
0x41FF_F060	Reserved
0x41FF_F061	
0x41FF_F062	
0x41FF_F063	
0x41FF_F064	Reserved
0x41FF_F065	
0x41FF_F066	
0x41FF_F067	
0x41FF_F068	Reserved
0x41FF_F069	
0x41FF_F06A	
0x41FF_F06B	
0x41FF_F06C	Reserved
0x41FF_F06D	
0x41FF_F06E	
0x41FF_F06F	

Address	Register name
0x41FF_F070	Reserved
0x41FF_F071	
0x41FF_F072	
0x41FF_F073	
0x41FF_F074	Reserved
0x41FF_F075	
0x41FF_F076	
0x41FF_F077	
0x41FF_F078	Reserved
0x41FF_F079	
0x41FF_F07A	
0x41FF_F07B	
0x41FF_F07C	Reserved
0x41FF_F07D	
0x41FF_F07E	
0x41FF_F07F	

Address	Register name
0x41FF_F080	Reserved
0x41FF_F081	
0x41FF_F082	
0x41FF_F083	
0x41FF_F084	Reserved
0x41FF_F085	
0x41FF_F086	
0x41FF_F087	
0x41FF_F088	Reserved
0x41FF_F089	
0x41FF_F08A	
0x41FF_F08B	
0x41FF_F08C	Reserved
0x41FF_F08D	
0x41FF_F08E	
0x41FF_F08F	

Address	Register name
0x41FF_F090	Reserved
0x41FF_F091	
0x41FF_F092	
0x41FF_F093	
0x41FF_F094	Reserved
0x41FF_F095	
0x41FF_F096	
0x41FF_F097	
0x41FF_F098	Reserved
0x41FF_F099	
0x41FF_F09A	
0x41FF_F09B	
0x41FF_F09C	Reserved
0x41FF_F09D	
0x41FF_F09E	
0x41FF_F09F	

Address	Register name
0x41FF_F0A0	Reserved
0x41FF_F0A1	
0x41FF_F0A2	
0x41FF_F0A3	
0x41FF_F0A4	Reserved
0x41FF_F0A5	
0x41FF_F0A6	
0x41FF_F0A7	
0x41FF_F0A8	Reserved
0x41FF_F0A9	
0x41FF_F0AA	
0x41FF_F0AB	
0x41FF_F0AC	Reserved
0x41FF_F0AD	
0x41FF_F0AE	
0x41FF_F0AF	

Address	Register name
0x41FF_F0B0	Reserved
0x41FF_F0B1	
0x41FF_F0B2	
0x41FF_F0B3	
0x41FF_F0B4	Reserved
0x41FF_F0B5	
0x41FF_F0B6	
0x41FF_F0B7	
0x41FF_F0B8	Reserved
0x41FF_F0B9	
0x41FF_F0BA	
0x41FF_F0BB	
0x41FF_F0BC	Reserved
0x41FF_F0BD	
0x41FF_F0BE	
0x41FF_F0BF	

26 Port Section Equivalent Circuit Schematics

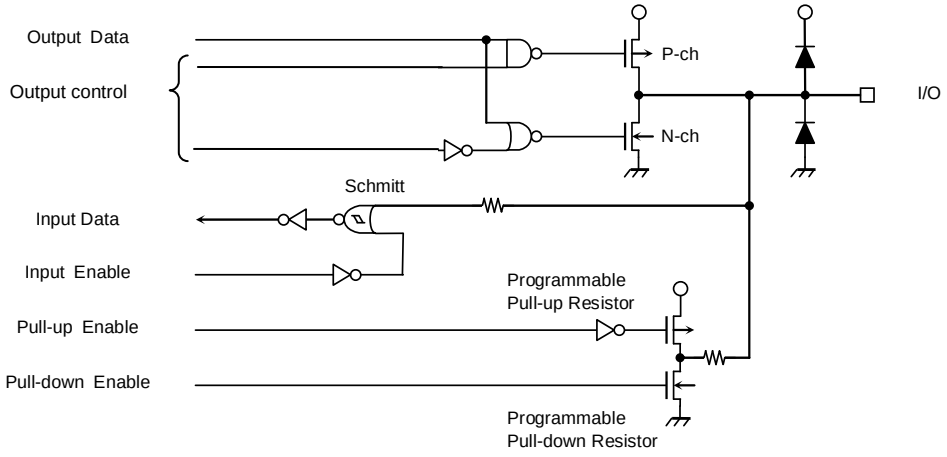
Important
TMPM382 (64-pin version) does not have PD0-6,PE6,PE7,PF2-4,PG0-7,PJ0-7,PN0-7 total 36 pins.

- How to read the schematics

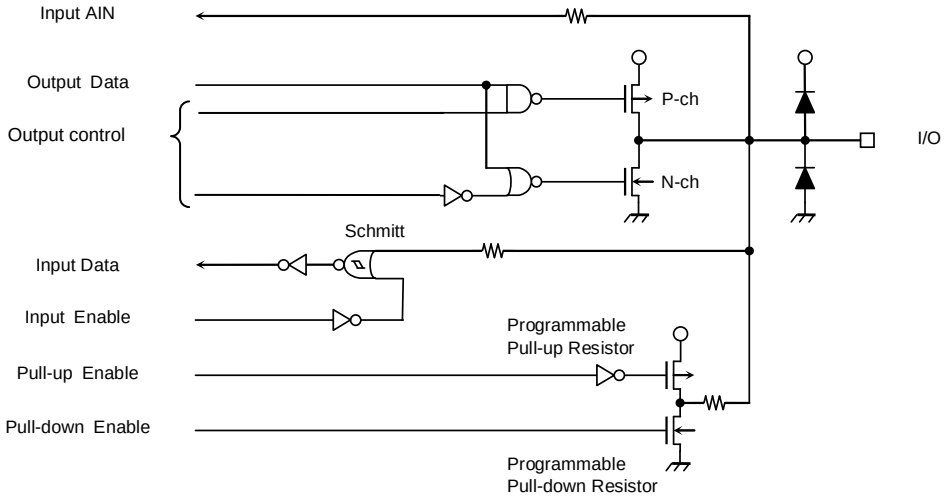
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The input protection resistance ranges from several tens of ohms to several hundreds of ohms. Damping resistor and Feedback resistor are shown with a typical value.

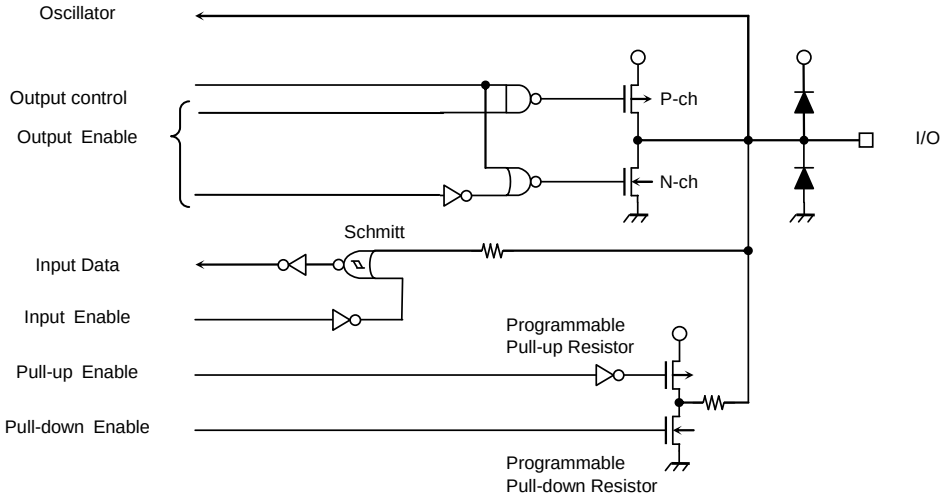
- PA0-7, PB0-7, PC0-7, PD0-6, PE0-7, PF0-4, PG0-7, PL2, PN0-7



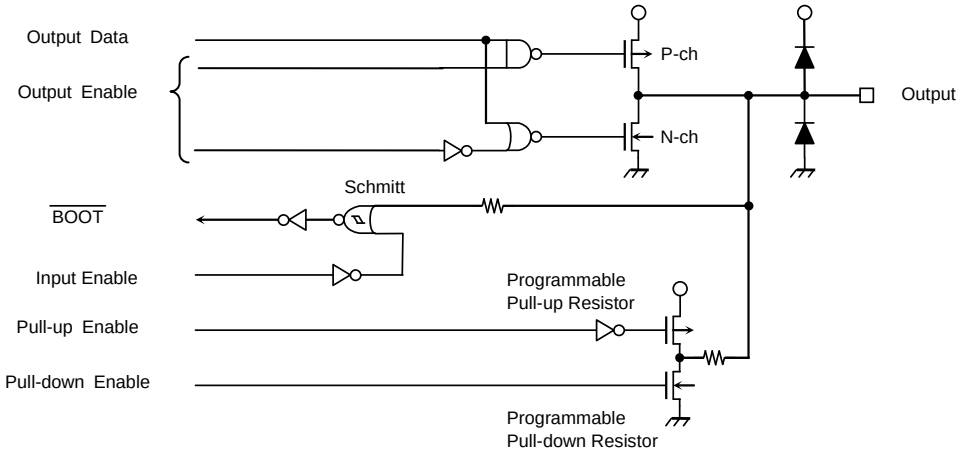
- PH0-7, PI0-1, PJ0-7



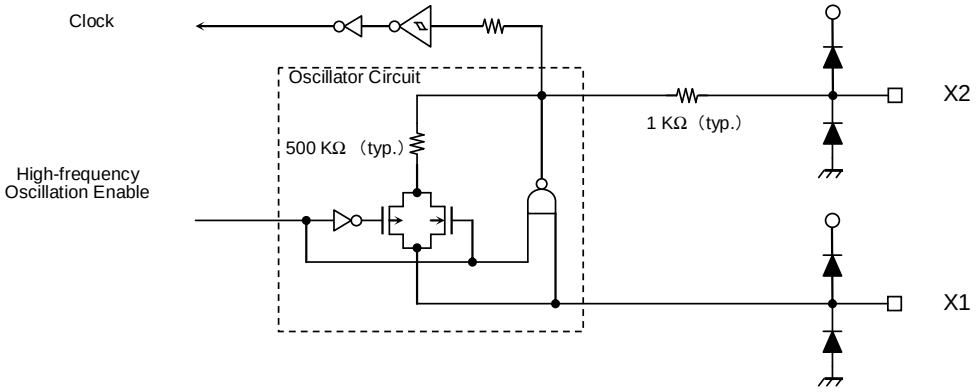
- PM0-1, PP0-1



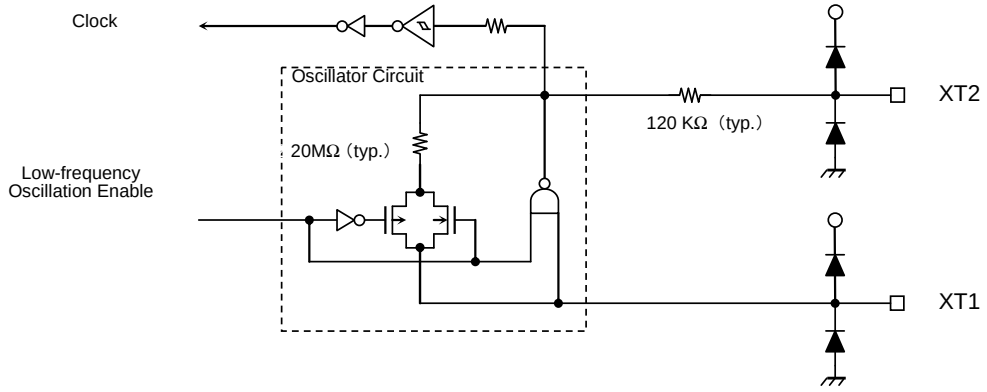
- PL0



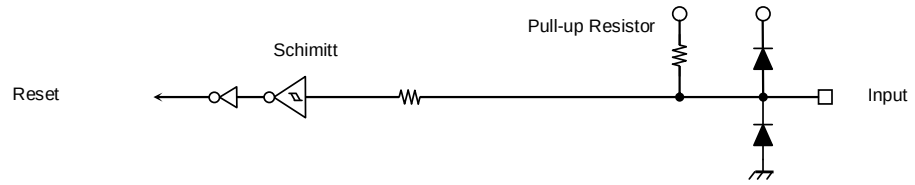
- X1, X2



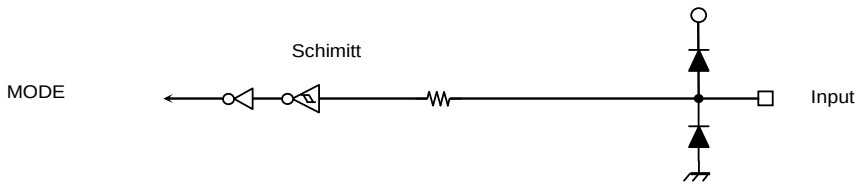
- XT1, XT2



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

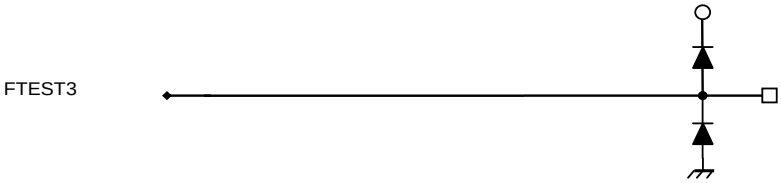


- MODE



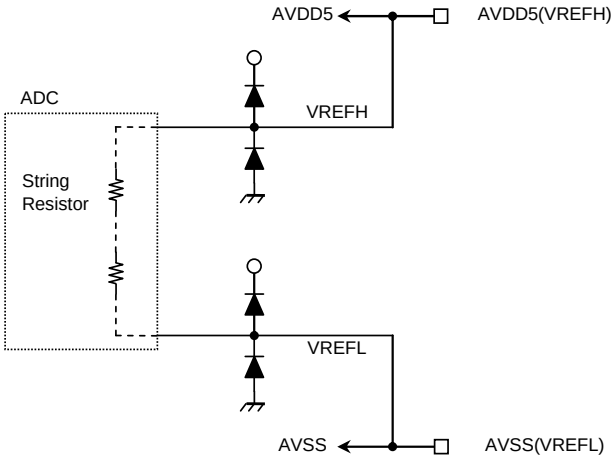
Note : MODE must be connected with GND.

- FTEST3



Note : FTEST3 must be OPEN.

- AVDD5(VREFH), AVSS(VREFL)



27 Electrical Characteristics

27.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		DVDD5	− 0.3 to 6.0	V
		RVDD5	− 0.3 to 6.0	
		AVDD5	− 0.3 to 6.0	
Input voltage		VIN	− 0.3 to VDD+0.3	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 (10s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	− 55 to 125	°C
Operating Temperature	Except during Flash W/E	T_{OPR}	− 40 to 85	°C
	During Flash W/E		0 to 70	

(Note) Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

27.2 DC Electrical Characteristics (1/3)

Ta= -40 to 85°C

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Supply voltage (Note 2)	DVDD5 AVDD5 RVDD5	DVDD5 = RVDD5 =AVDD5 DVSS = AVSS = 0V fosc = 8 ~ 10MHz fsys = 1 ~ 40MHz fs = 30 ~ 34KHz	4.0	−	5.5	V
Capacitance for VOUT3 (Note 3)	Cout	DVDD5=4.0 to 5.5V	3.3	−	4.7	μF
Low-level input voltage	VIL1	DVDD5=4.0 to 5.5V (PORT A/B/C/D/E/F/G/L/M/N/P)	-0.3	−	0.25 DVDD5	V
	VIL2	AVDD5=4.0 to 5.5V (PORT H/I/J)	-0.3	−	0.25 AVDD5	V
High-level input voltage	VIH1	DVDD5=4.0 to 5.5V (PORT A/B/C/D/E/F/G/L/M/N/P)	0.75 DVDD5	−	DVDD5 ＋ 0.3	V
	VIH2	AVDD5=4.0 to 5.5V (PORT H/I/J)	0.75 AVDD5	−	AVDD5 ＋ 0.3	V
Low-level output voltage	VOL1	DVDD5≧4.0V ／ IOL = 1.6mA (PORT A/B/C/D/E/F/G/L/M/N/P)	−	−	0.4	V
	VOL2	AVDD5≧4.0V ／ IOL = 1.6mA (PORT H/I/J)	−	−	0.4	V
High-level output voltage	VOH1	DVDD5≧4.0V ／ IOH=–1.6mA (PORT A/B/C/D/E/F/G/L/M/N/P)	DVDD5 −0.4	−	−	V
	VOH2	AVDD5≧4.0V ／ IOH=–1.6mA (PORT H/I/J)	AVDD5 −0.4	−	−	V
Input leakage current	ILI	0.0V ≦ VIN ≦DVDD5 0.0V ≦ VIN ≦AVDD5	−	0.02	± 5	μA
Output leakage current	ILO	0.2V ≦ VIN ≦DVDD 5– 0.2V 0.2V ≦ VIN ≦AVDD5 – 0.2V	−	0.05	± 10	μA
Pull-up resister at Reset	RRST	4.0V≦DVDD5≦5.5V	38.5	50	71.4	kΩ
Schmitt-Triggered port	VTH	4.0V≦DVDD5≦5.5V 4.0V≦AVDD5≦5.5V	0.3	0.6	−	V
Programmable pull-up/ pull-down resistor	PKH	4.0V≦DVDD5≦5.5V 4.0V≦AVDD5≦5.5V	38.5	50	71.4	kΩ
Pin capacitance (Except power supply pins)	C_{IO}	fc = 1MHz	−	−	10	pF

(Note 1) Ta = 25°C, DVDD5 = AVDD5 = RVDD5 = 5V, unless otherwise noted.

(Note 2) The same voltage must be supplied to DVDD5, AVDD5 and RVDD5.

(Note 3) VOUT3 pin should be connected to GND via a capacitance.

27.3 DC Electrical Characteristics (2/3)

Ta＝－40 to 85℃, DVDD5=RVDD5=AVDD5=4.0 to 5.5V

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Low-level output Current	IOL	Per pin	－	－	2	mA
	ΣIOL1	Per group @ 4.0V≦DVDD5≦5.5V GrL1 = <PA0-7/PE0-5/PG0-7> GrL2 = <PB0-7/PD0-6/PF0-4/PL0> GrL3 = <PC0-7/PM0-1/PP0-1> GrL4 = <PE6-7/PL2/PN0-7>	－	－	20	mA
	ΣIOL2	Per group @ 4.0V≦AVDD5≦5.5V GrL5 = <PH0-7/PI0-1/PJ0-7>	－	－	9	mA
	ΣIOL	Total of all pins	－	－	30	mA
High-level output Current	IOH	Per pin	－	－	-2	mA
	ΣIOH1	Per groups @ 4.0V≦DVDD5≦5.5V GrH1 = <PA0-7/PE0-3/PG0-7/PM0-1/PP0-1> GrH2 = <PB0-7/PC0-7/PD0-6/PF0-4/PL0> GrH3 = <PE4-7/PL2/PN0-7>	－	－	-20	mA
	ΣIOH2	Per groups @ 4.0V≦AVDD5≦5.5V GrH4 = <PH0-7/PI0-1/PJ0-7>	－	－	-9	mA
	ΣIOH	Total of all pins	－	－	-30	mA

(Note) Current can flow capacity in each condition.

27.4 DC Electrical Characteristics (3/3)

Ta＝－40 to 85℃, DVDD5=RVDD5=AVDD5=4.0 to 5.5V

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
NORMAL (Note 2)	I_{DD}	fsys＝40MHz (fosc＝10MHz, Gear 1/1)	－	25	33	mA
IDLE (Note 3)			－	19	26	
SLOW		fs=32.768kHz	－	382	1850	μA
SLEEP (Note 4)			－	122	800	μA
STOP			－	96	750	μA
FLASH Write Erase Current			－	25	35	mA

(Note 1) Ta = 25°C, DVDD5 = AVDD5 = RVDD5 = 5V, unless otherwise noted.

(Note 2) I_{DD} NORMAL: Measured with the dhrystone ver. 2.1 operated in FLASH. All functions operates excluding A/D.

(Note 3) I_{DD} IDLE:Measured with

CPU is stopped, some of the periperal is running.

(Note 4) I_{DD} SLEEP:Measured with

CPU is stopped using RMC、RTC only.

27.5 12-bit ADC Electrical Characteristics

Ta＝－40～85℃, DVDD5=RVDD5＝4.5V～5.5V, DVSS=AVSS=0V

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog supply voltage（Note1）	AVDD5	AVDD5=V_{REFH}	DVDD5 －0.2	－	DVDD5	V
Analog reference voltage	AVSS	AVSS=V_{REFL}	0	－	0	V
Analog input voltage	V_{AIN}	－	AVSS	－	AVDD5	V
Analog reference supply current（Note4）	I_{REF}	IREF ON (During AD conversion)	－	7.5	10.0	mA
		IREF ON(During AD stop)	－	3.5	5	mA
		IREF OFF(During STOP MODE)	－	3	70	μA
INL error	－	12bit mode AIN resistance ≦600Ω AIN load capacitance ≦0.1μF Conversion time ≧1.85μs	－	－	±9	LSB (Note2)
DNL error			－	－	+6～－1	
Offset error			－	－	±5	
Full-scale error			－	－	+8～－2	
Total error			－	－	+12～－8	
INL error	－	10bit mode AIN resistance ≦600Ω AIN load capacitance ≦0.1μF Conversion time ≧1.70μs	－	－	±3	LSB (Note3)
DNL error			－	－	±2	
Offset error			－	－	±3	
Full-scale error			－	－	±3	
Total error			－	－	±4	

(Note 1) A/D when using separate power supply for the converter, you must keep this condition.

(Note 2) 1LSB = (AVDD5 – AVSS) / 4096[V]

(Note 3) 1LSB = (AVDD5 – AVSS) / 1024[V]

(Note 4) The relevant pin for I_{REF} is AVDD5, so that the current flowing into AVDD5 is the power supply current AVDD5 + I_{REF} .

(Note) Peripheral functions are disable.

27.6 AC Electrical Characteristics

27.6.1 AC measurement condision

The AC characteristics data of this chapter is measured under the following conditions unless otherwize noted.

- 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
- Ta : -40 to 85 ℃

(Note)The "Equation" column in the table shows the specifications under the conditions DVDD5 = 4.0 to 5.5 V.

27.6.2 Serial Channel Timing (UART/SIO)

(1) I/O Interface mode

In the table below, the letter x represents the SIO operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

1) SCLK input mode

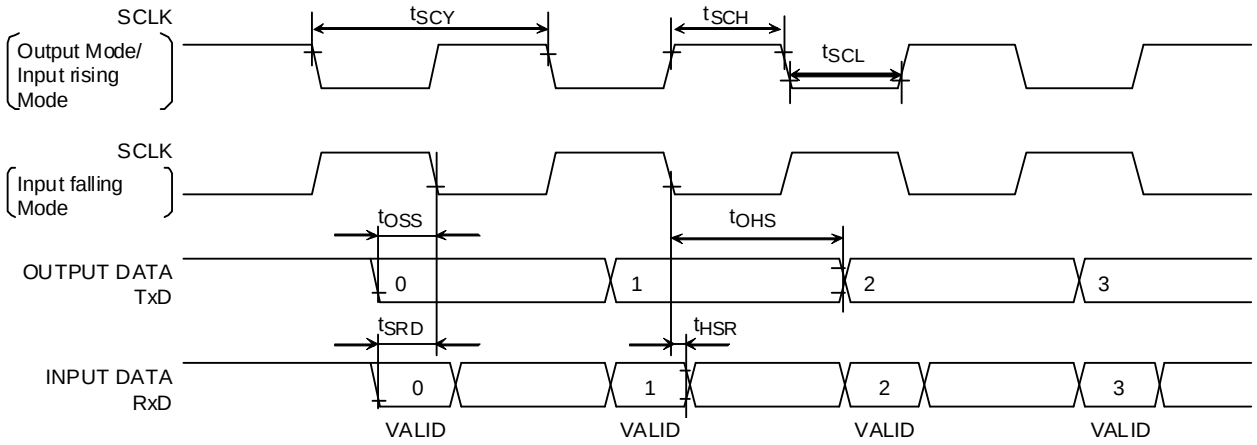
Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK Clock High width (input)	t_{SCH}	3x	-	75	-	ns
SCLK Clock Low width (input)	t_{SCL}	3x	-	75	-	
SCLK cycle	t_{SCY}	$t_{SCH} + t_{SCL}$	-	150	-	
TxD to SCLK rise or fall (Note 1)	t_{OSS}	t_{SCY} /2 – 3x - 45	-	-45 (Note 2)	-	
TxD hold or fall after SCLK rising (Note 1)	t_{OHS}	t_{SCY} /2	-	75	-	
RxD valid to SCLK rise or fall (Note 1)	t_{SRD}	30	-	30	-	
RxD hold or fall after SCLK rising (Note 1)	t_{HSR}	x + 30	-	55	-	

(Note 1) SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

(Note 2) Keep this value positive by adjusting SCLK cycle.

2) SCLK output mode

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	4x	-	100	-	ns
Output Data ← SCLK rise	t_{OSS}	t_{SCY} /2 - 20	-	30	-	
SCLK rise → Output hold Data hold	t_{OHS}	t_{SCY} /2 - 20	-	30	-	
Valid Data Input ← SCLK rise	t_{SRD}	45	-	45	-	
SCLK rise → Input Data hold	t_{HSR}	0	-	0	-	



27.6.3 Serial Bus Interface（I2C/SIO）

(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 SBInCR.

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, 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) is calculated with: $(2^{n-1}+58)/x$

(Note 2) SCL clock high width (output) is calculated with: $(2^{n-1}+12)/x$

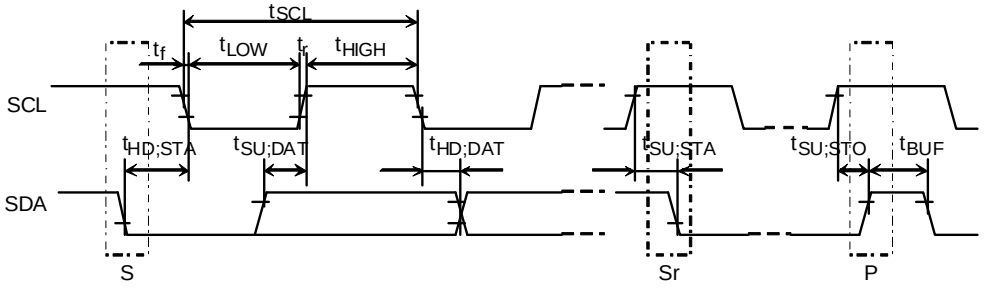
Notice: On I2C-bus specification, Maximum Speed of Standard Mode is 100KHz, Fast mode is 400KHz. Internal SCL Frequency setting should comply with Note1 & Note2 shown above.

(Note 3) The output data hold time is equal to 12x 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: Repeated start condition
P: Stop condition

(2) Clock-Synchronous 8-Bit SIO mode

In the table below, the letter x represents the I2C operation clock cycle time which is identical to the fsys cycle time. It varies depending on the programming of the clock gear function.

The electrical specifications below are for an SCK signal with a 50% duty cycle.

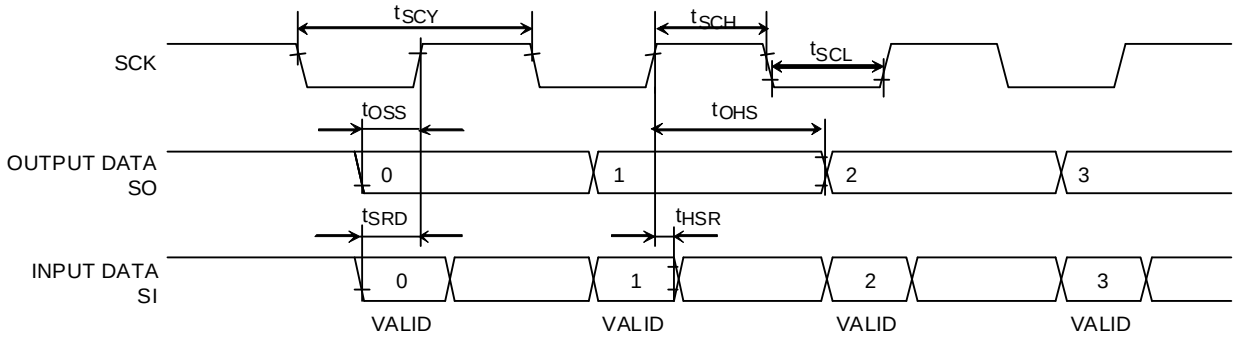
1) SCK Input Mode

Parameter	Symbol	Equation		40MHz		Unit
		Min	Max	Min	Max	
SCK Clock High width (input)	t_{SCH}	4x	-	100	-	ns
SCK Clock Low width (input)	t_{SCL}	4x	-	100.	-	
SCK cycle	t_{SCY}	8x	-	200	-	
Output Data ← SCK rise	t_{OSS}	t_{SCY}/2 - 3x - 45	-	-20 (Note)	-	
SCK rise → Output Data hold	t_{OHS}	t_{SCY}/2 + 2x - 25	-	125	-	
Valid Data input ← SCK rise	t_{SRD}	30 - x	-	5	-	
SCK rise → Input Data hold	t_{HSR}	30	-	30	-	

(Note) Keep this value positive by adjusting SCK cycle.

2) SCK Output Mode

Parameter	Symbol	Equation		40MHz		Unit
		Min	Max	Min	Max	
SCK cycle (programmable)	t_{SCY}	16x	-	400	-	ns
Output Data ← SCK rise	t_{OSS}	t_{SCY}/2 – 20	-	180	-	
SCK rise → Output Data hold	t_{OHS}	t_{SCY}/2 – 20	-	180	-	
Valid Data input ← SCK rise	t_{SRD}	45	-	45	-	
SCK rise → Input Data hold	t_{HSR}	0	-	0	-	



27.6.4 SPP Controllor (SSP)

AC measurement conditions

- The letter "T" used in the equation in the table represents the period of internal bus frequency(fpcLK)
- Output level: High=0.7×DVDD5, Low =0.3×DVDD5
- Input level: High=0.9×DVDD5, Low =0.1×DVDD5
- Load capacitance CL = 30 pF
- Ta = -45 to 85℃

(Note)The "Equation" column in the table shows the specifications under the conditions DVDD5 = 4.0 to 5.5 V.

Parameter	Symbol	Equation		fsys 40MHz (m=4 n=12)	Unit
		Min	Max		
SPxCLK Period (Master)	T_m	(m)T However more than、100nS		100 (10MHz)	nS
SPxCLK Period (Slave)	T_s	(n)T		300 (3.3MHz)	
SPxCLK rise up time	t_r		15.0	15.0	
SPxCLK fall down time	t_f		15.0	15.0	
Master mode: SPxCLK low level pulse width	t_{WLM}	(m)T / 2 - 20.0		30	
Master mode: SPxCLK high level pulse width	t_{WHM}	(m)T / 2 –20.0		30	
Slave mode: SPxCLK low level pulse width	t_{WLS}	(n)T / 2 –10.0		145	
Slave mode: SPxCLK high level pulse width	t_{WHS}	(n)T / 2 –10.0		145	
Master Mode: SPxCLK rise/fall to output data valid	t_{ODSM}		15.0	15.0	
Master Mode: SPxCLK rise/fall to output data hold	t_{ODHM}	(m)T/2 -15		35.0	
Master Mode: SPxCLK rise/fall to input data valid delay time	t_{IDSM}	35.0		35.0	
Master Mode: SPxCLK rise/fall to input data hold	t_{IDHM}	5.0		5.0	
Master Mode: SPxFSS valid to SPxCLK rise/fall	t_{OFSM}	(m)T -15	(m)T+15	85 – 115	
Slave mode: SPxCLK rise/fall to output data valid delay time	t_{ODSS}		(3T) + 35	110	
Slave mode: SPxCLK rise/fall to output data hold	t_{ODHS}	(n)T /2 ＋ (2T)		200	
Slave mode: SPxCLK rise/fall to input data valid delay time	t_{IDSS}	10		10	
Slave mode: SPxCLK rise/fall to input data hold	t_{IDHS}	(3T) +15		90	
Slave mode: SPxFSS valid to SPxCLK rise/fall	t_{OFSS}	(n)T -20		280	

(Note) Baud rate Clcok is set under below condition

Master mode

m = (<CPSDVSR> × (1+<SCR>)) = f_{sys} ／ SPxCLK

<CPSDVR> is set only even number and "m" must set during 65204 ≧ m ≧ 2

Slave mode

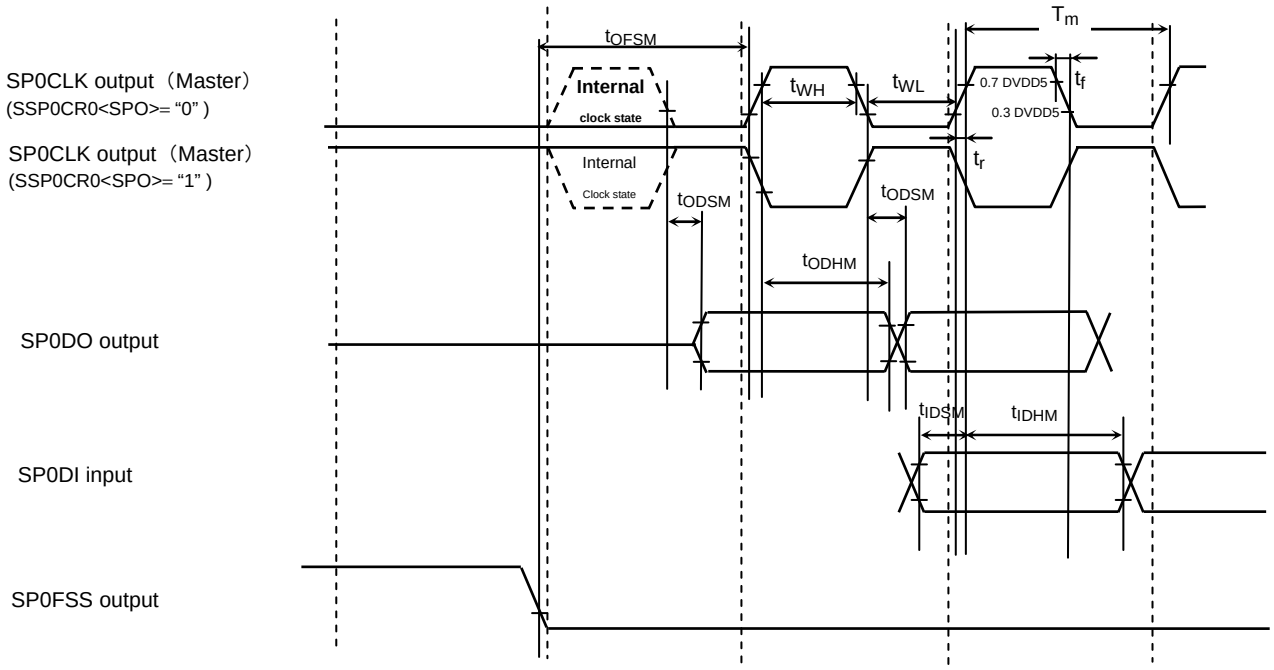
n = f_{sys} ／ SPxCLK (65024 ≧ n ≧ 12)

SSP SPI mode（Master）

f_{sys} ≧ 2 × SPxCLK(Max)

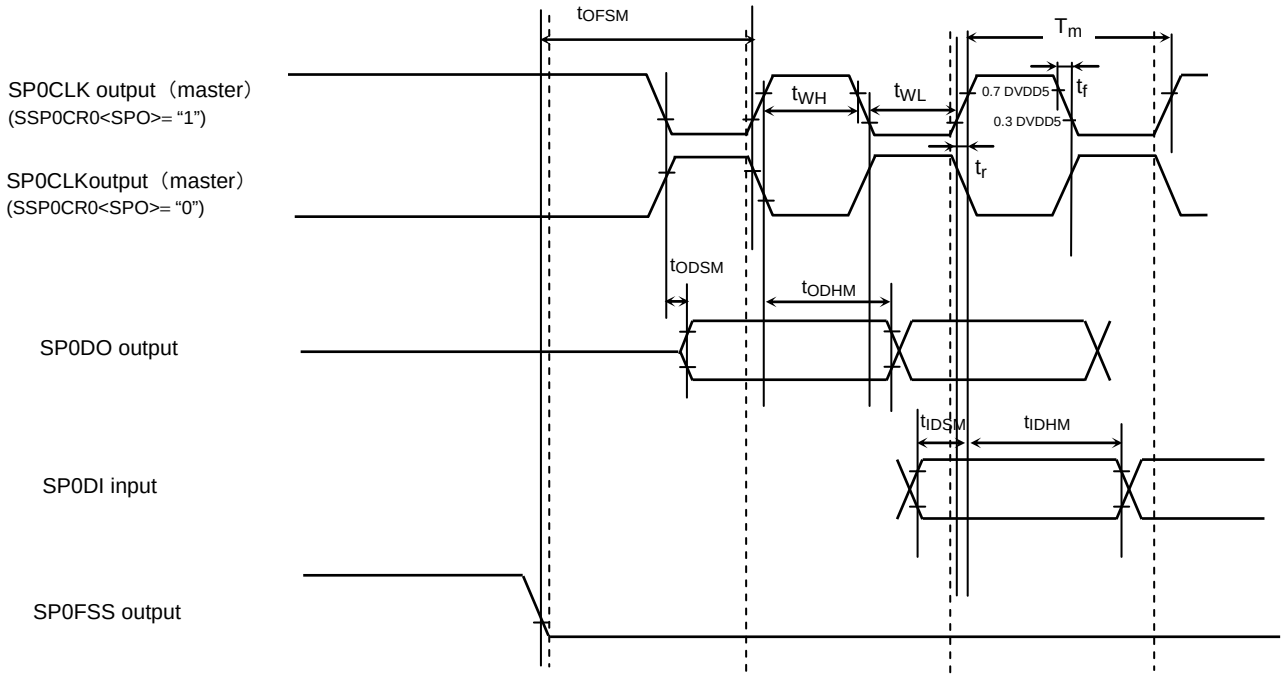
f_{sys} ≧ 65024 × SPxCLK(min)

(1) Master SSP0CR0<SPH>= “0”(Data is latched on the first edge)



SSP SPI mode（Master）

(2) Master SSP0CR0<SPH>= “1” (Data is latched on the second edge))

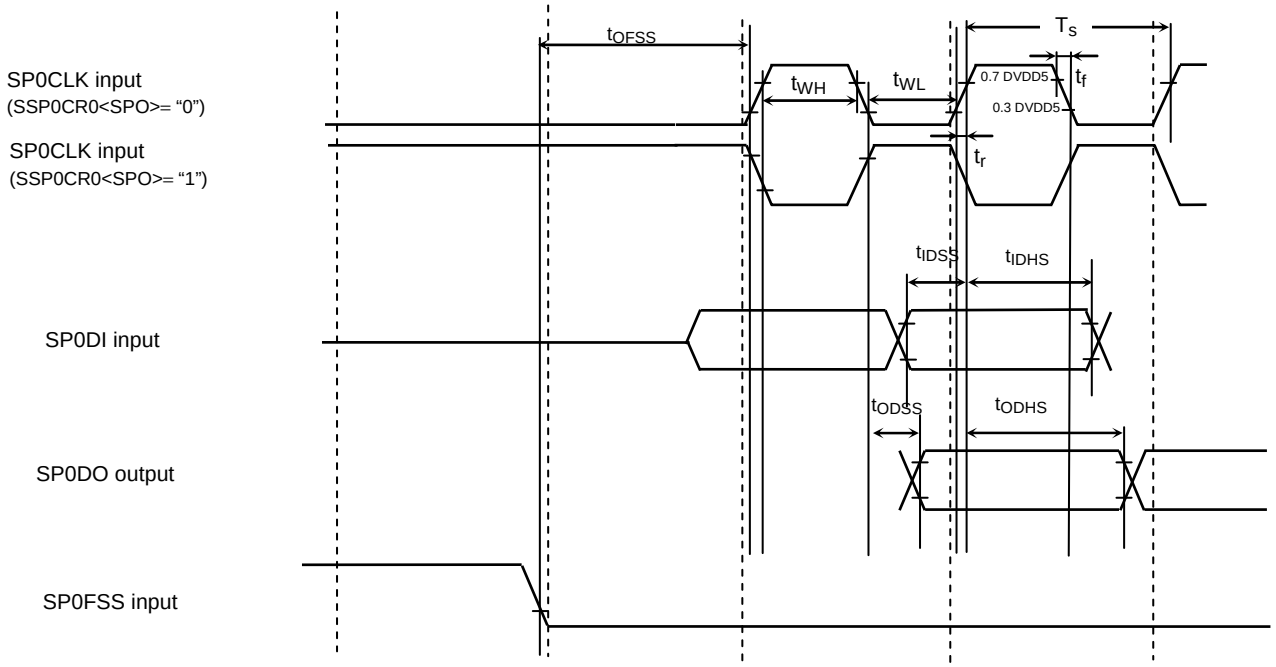


SSP SPI mode(Slave)

$f_{sys} \geqq 12 \times$ SPxCLK (max)

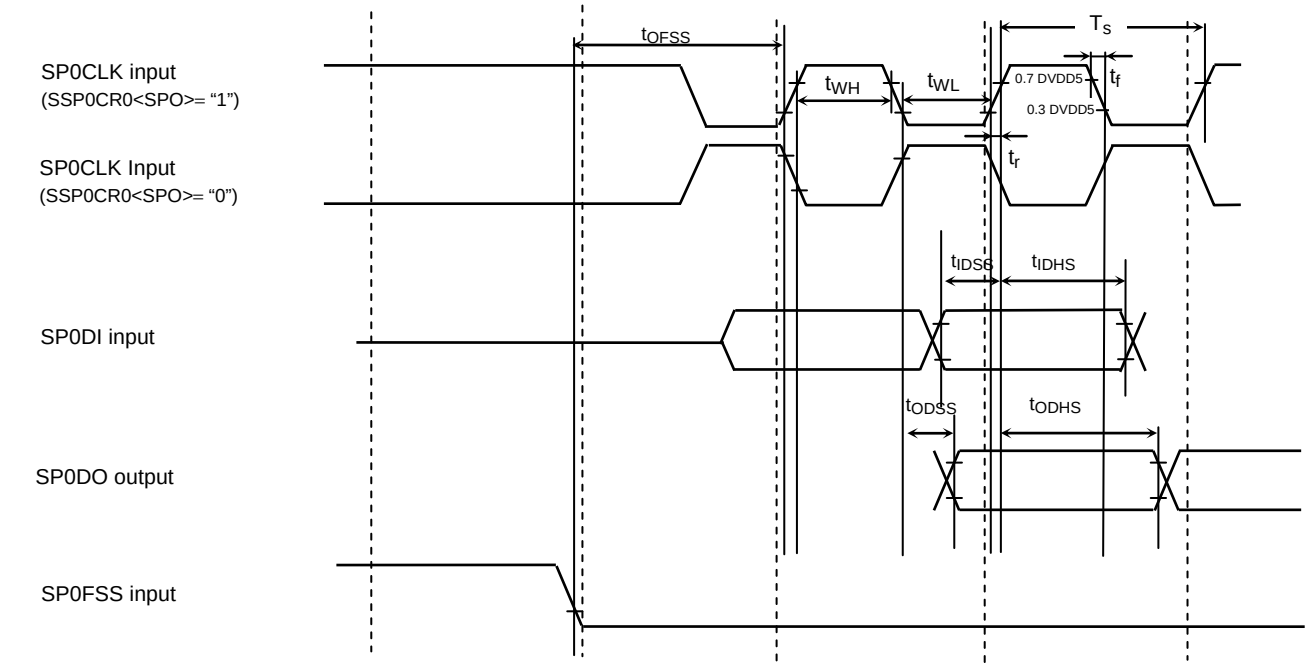
$f_{sys} \geqq 65024 \times$ SPxCLK (min)

(3) Slave SSP0CR0<SPH>= “0”(Data is latched on the first edge)



SSP SPI mode(Slave)

(4) Slave SSP0CR0<SPH>= “1” (Data is latched on the second edge))



27.6.5 Event Counter (TMRB)

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		40MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2x + 100	-	150	-	ns
Clock high pulse width	t_{VCKH}	2x + 100	-	150	-	ns

27.6.6 Capture (TMRB)

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		40MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2x + 100	-	150	-	ns
High pulse width	t_{CPH}	2x + 100	-	150	-	ns

27.6.7 General Interrupts (INT)

In the table below, the letter x represents the fsys cycle time.

Parameter	Symbol	Equation		40MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0 to INTF	t_{CPL}	x + 100	-	125	-	ns
High pulse width for INT0 to INTF	t_{CPH}	x + 100	-	125	-	ns

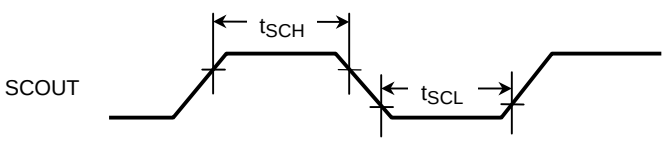
27.6.8 STOP Release Interrupts

Parameter	Symbol	Equation		40MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0 to INTF	t_{INTBL}	100	-	100	-	ns
High pulse width for INT0 to INTF	t_{INTBH}	100	-	100	-	ns

27.6.9 SCOUT Pin AC Characteristic

Parameter	Symbol	Equation		40MHz		Unit
		Min	Max	Min	Max	
High pulse width	t_{SCH}	0.5T - 5		7.5		ns
Low pulse width	t_{SCL}	0.5T - 5		7.5		ns

(Note) In the above table, the letter T represents the cycle time of the SCOUT output clock.



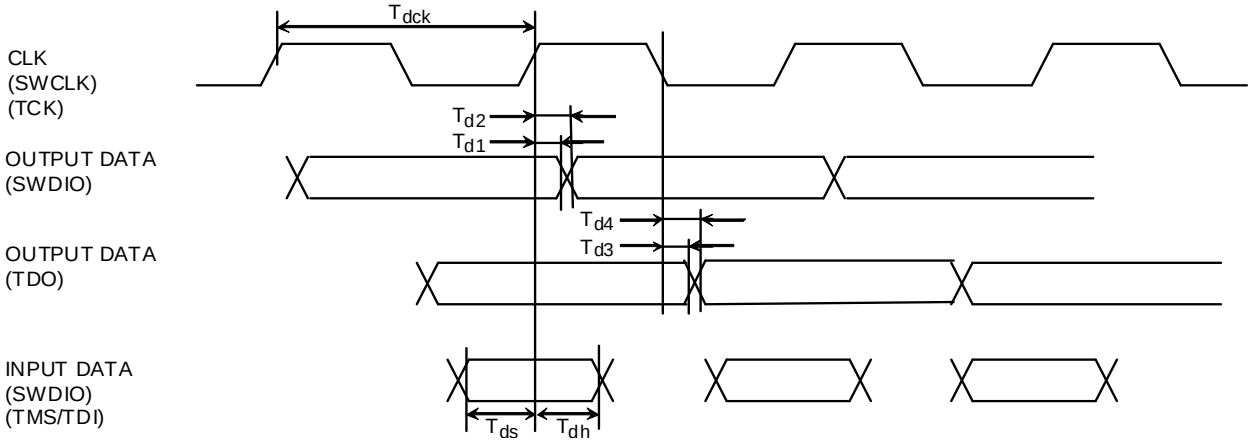
27.6.10 Debug Communication

(1) SWD Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	Tdck	100	-	ns
CLK rise → Output data hold	Td1	4	-	ns
CLK fall → Output data hold	Td2	-	37	ns
CLK fall → Output data hold	Tds	20	-	ns
CLK rise → Input data hold	Tdh	15	-	ns

(2) JTAG Interface

Parameter	Symbol	Min	Max	Unit
CLK cycle	Tdck	100	-	ns
CLK rise → Output data hold	Td3	4	-	ns
CLK fall → Output data hold	Td4	-	37	ns
Input data valid ← CLK rise	Tds	20	-	ns
CLK rise → Input data hold	Tdh	15	-	ns

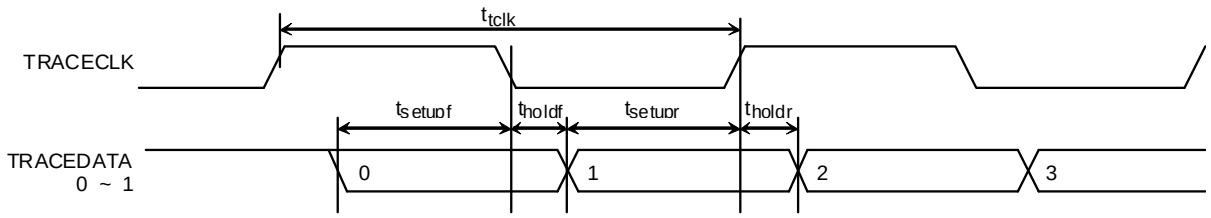


27.6.11 ETM Trace

AC measurement conditions

- 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}	50	-	ns
TRACEDATA valid ← TRACECLK rise	t_{setupr}	2	-	ns
TRACECLK rise → TRACEDATA hold	t_{holdr}	1	-	ns
TRACEDATA valid ← TRACECLK fall	t_{setupf}	2	-	ns
TRACECLK fall → TRACEDATA hold	t_{holdf}	1	-	ns



27.7 Flash characteristics

Ta＝0 to 70℃

Parameter	Condition	Min	Typ.	Max	Unit
Flash memory erase / write times				100	times

27.8 Internal Oscillator

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Oscillation frequency	fosc2	Ta＝25℃	-	9.0	-	MHz
Oscillation accuracy		Ta＝-40～85℃	-15	-	+15	%

27.9 Oscillation Circuit

Connection example

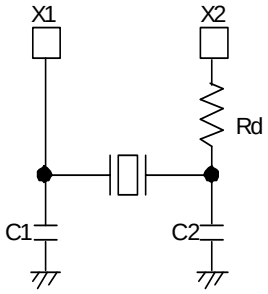


Fig 27.-1 High-frequency oscillation connection

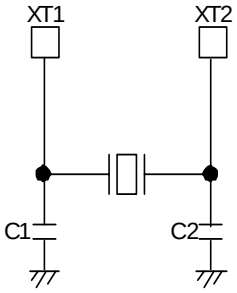


Fig 27-2 Low-frequency oscillation connection

(Note)The load value of the oscillator is the sum of loads (C1 and C2) and the floating load of the actual assembled board. There is a possibility of operating error when using C1 and C2 values in the table below. When designing the board, design the minimum length pattern around the oscillator. We also recommend that oscillator evaluation be carried out using the actual board.

The TX03 has been evaluated by the oscillator vender below. Use this information when selecting external parts.

27.9.1 Ceramic oscillator

The TX03 recommends the high-frequency oscillator by Murata Manufacturing Co., Ltd.

Please refer to the following URL for details.

http://www.murata.co.jp

27.9.2 Crystal oscillator

The TX03 recommends the high-frequency oscillator by KYOCERA KINSEKI Corporation.

Please refer to the following URL for details.

http://www.kinseki.co.jp

27.10 Handling Precaution

27.10.1 Solderability of test conditions

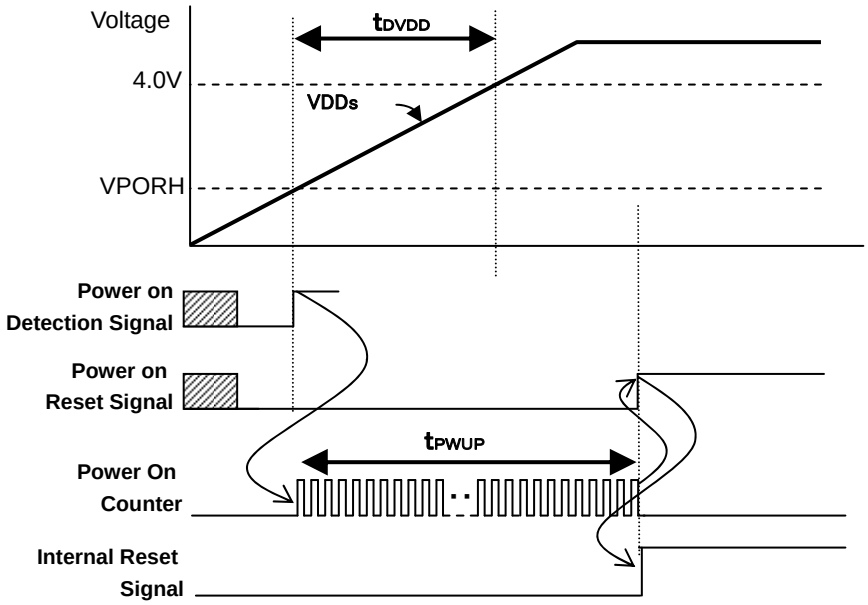
Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	

27.11 Note the power on

27.11.1 Using Power On Reset only

Table 27.1. Warming-up time and Rising time of power line (POR only)

Symbol	Rating	Min	Typ.	Max	Unit
t_{PWUP}	Warming-up time after reset released		$2^{13}/f_{osc}$		s
t_{DVDD}	Rising time of power line			0.6	ms

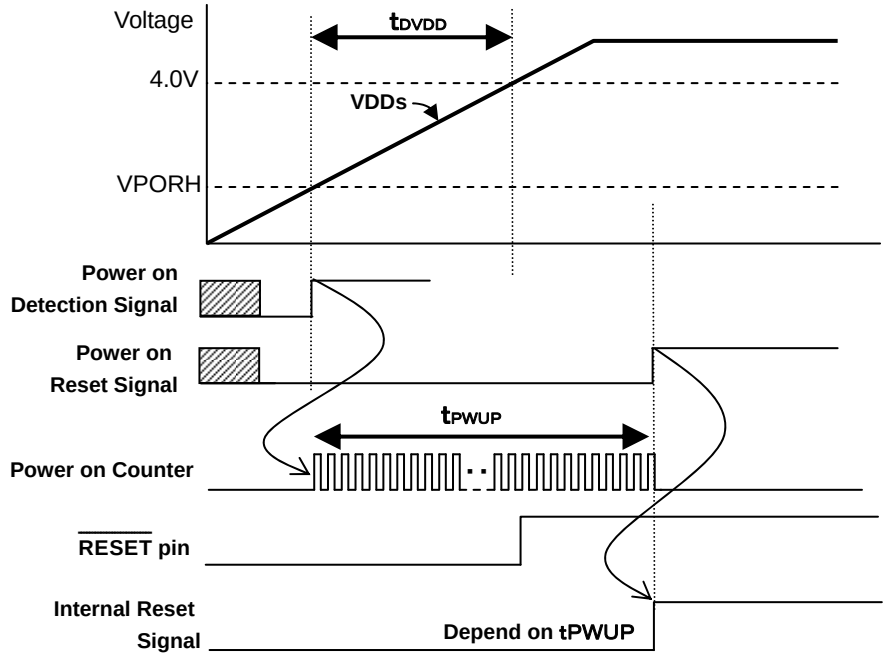


(Note) VDDs are DVDD5, RVDD5, AVDD5.

Fig 27-3 Power on Sequence (Using POR only)

27.11.2 Using External reset

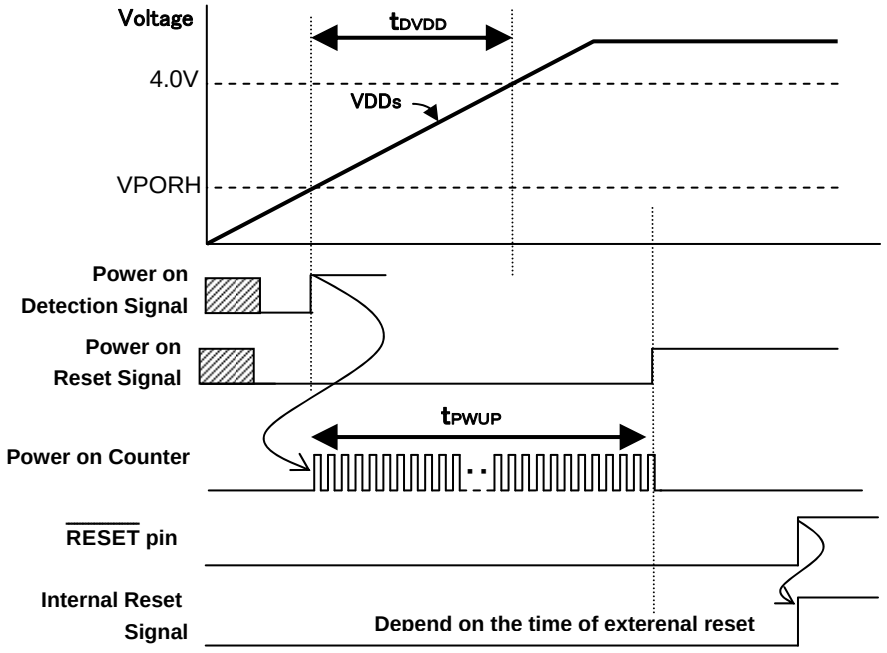
(1) In case of the time of external reset shorter than POR



(Note) VDDs are DVDD5, RVDD5, AVDD5.

Fig 27-4 Power on Sequence (Using POR and External reset)(1)

(2) In case of the time of external reset longer than tPWUP

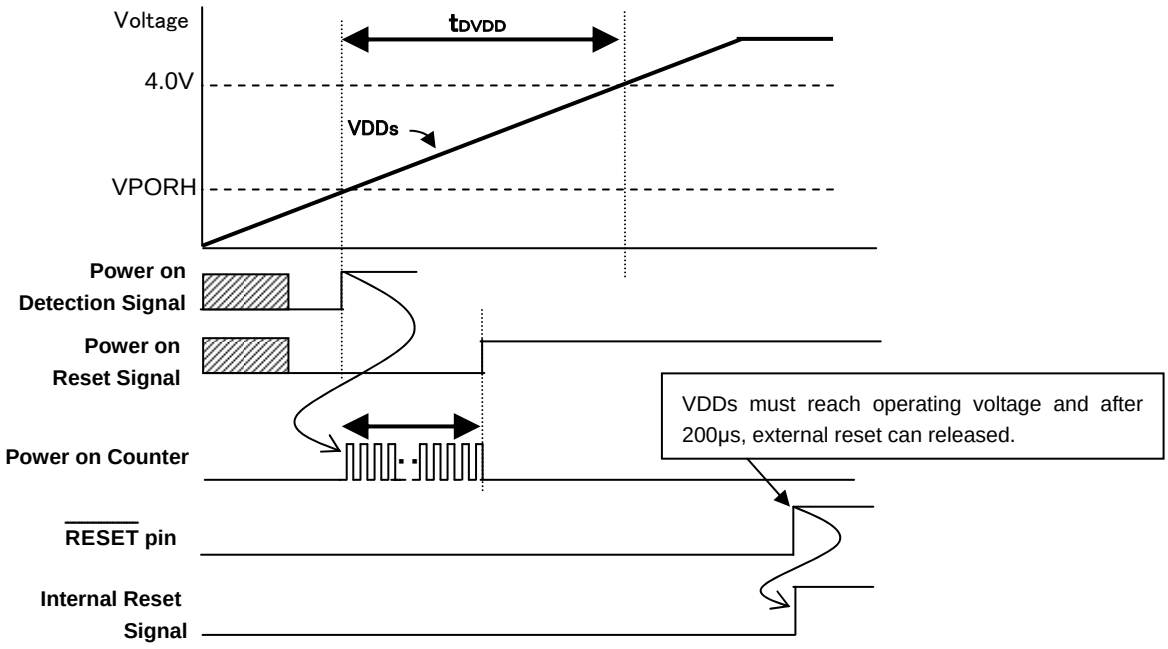


(Note) VDDs are DVDD5, RVDD5, AVDD5.

Fig 27-5 Power on Sequence (Using POR and External reset)(2)

(3) In case of the rising time of power line longer than tPWUP

(Note) In this case, must be reset from $\overline{\text{RESET}}$ pin.



(Note) VDDs are DVDD5, RVDD5, AVDD5.

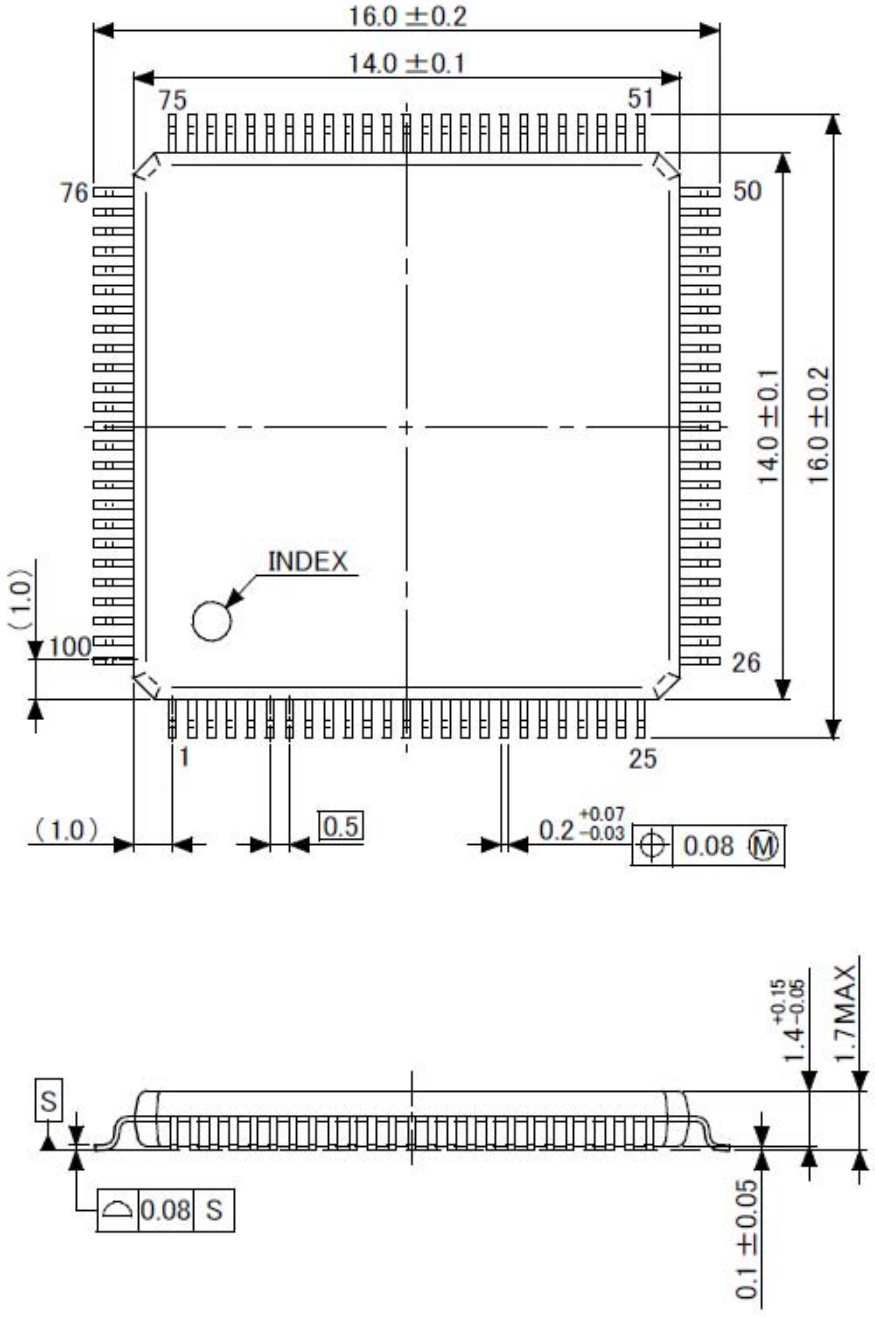
Fig 27-6 Power on Sequence (tDVDD > tPWUP)

28 Package

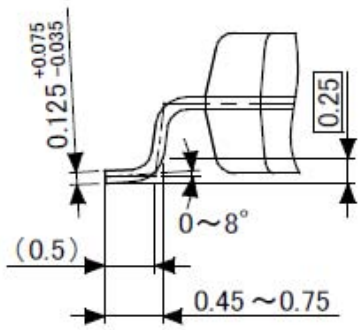
Type：LQFP100-P-1414-0.50H

Dimensions

Unit : mm



Pin detail

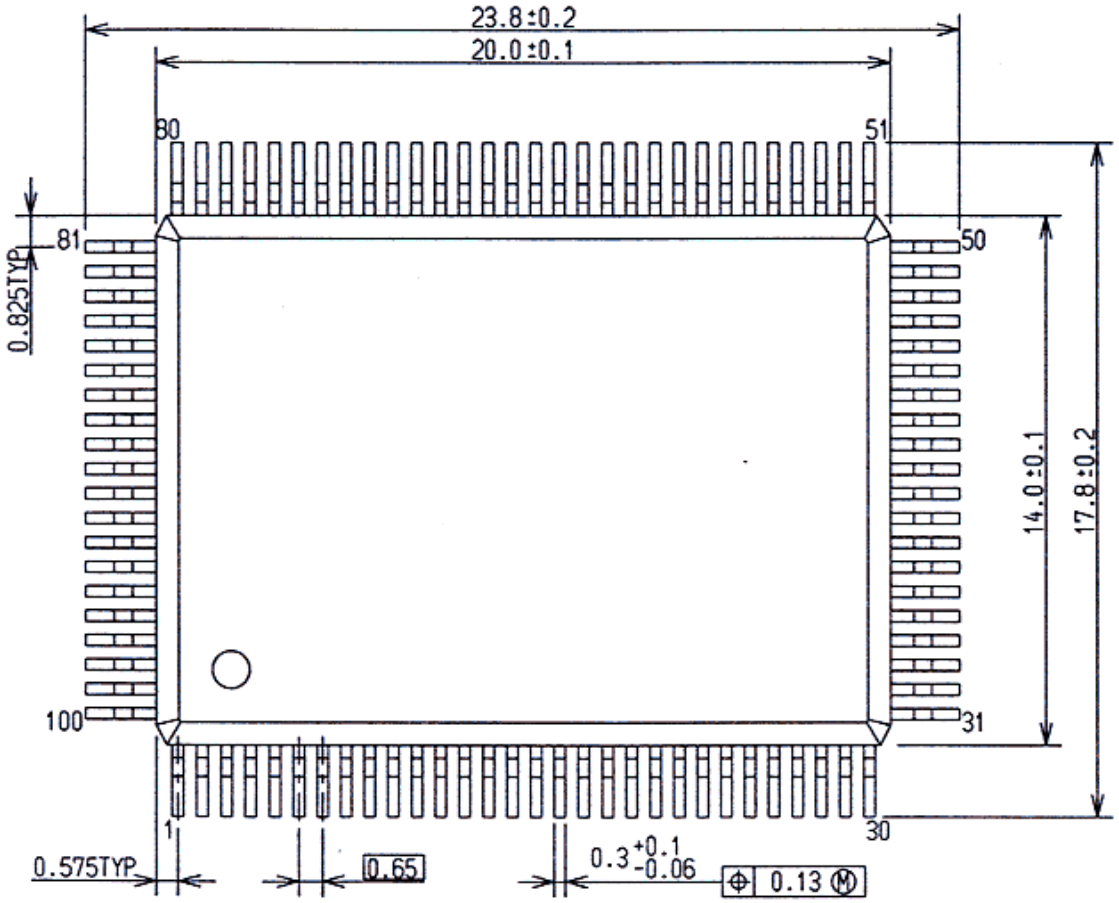


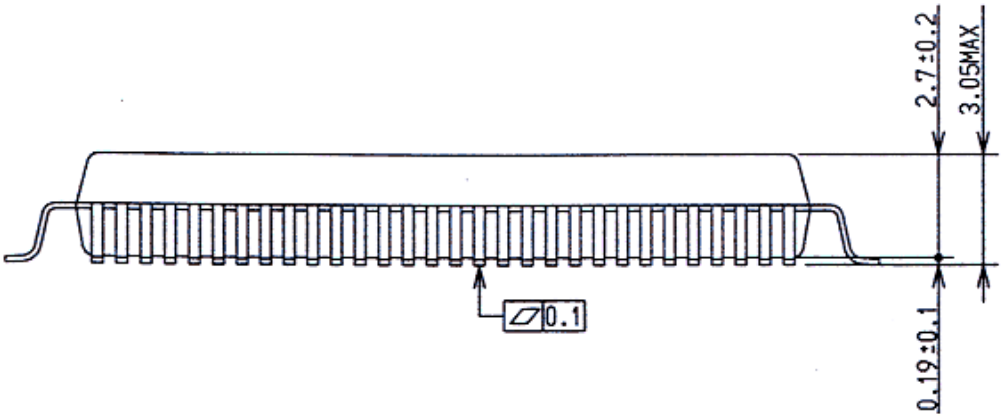
(Note 1)	**For more dimensional information, please contact any one of our representatives**
(Note 2)	**The package is palladized**

Type：QFP100-P-1420-0.65Q

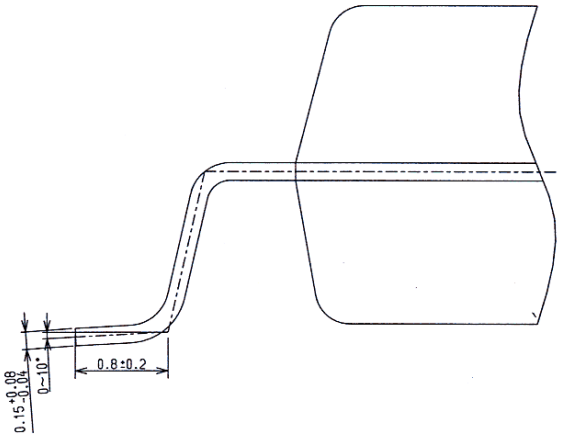
Dimensions

Unit : mm





Pin detail

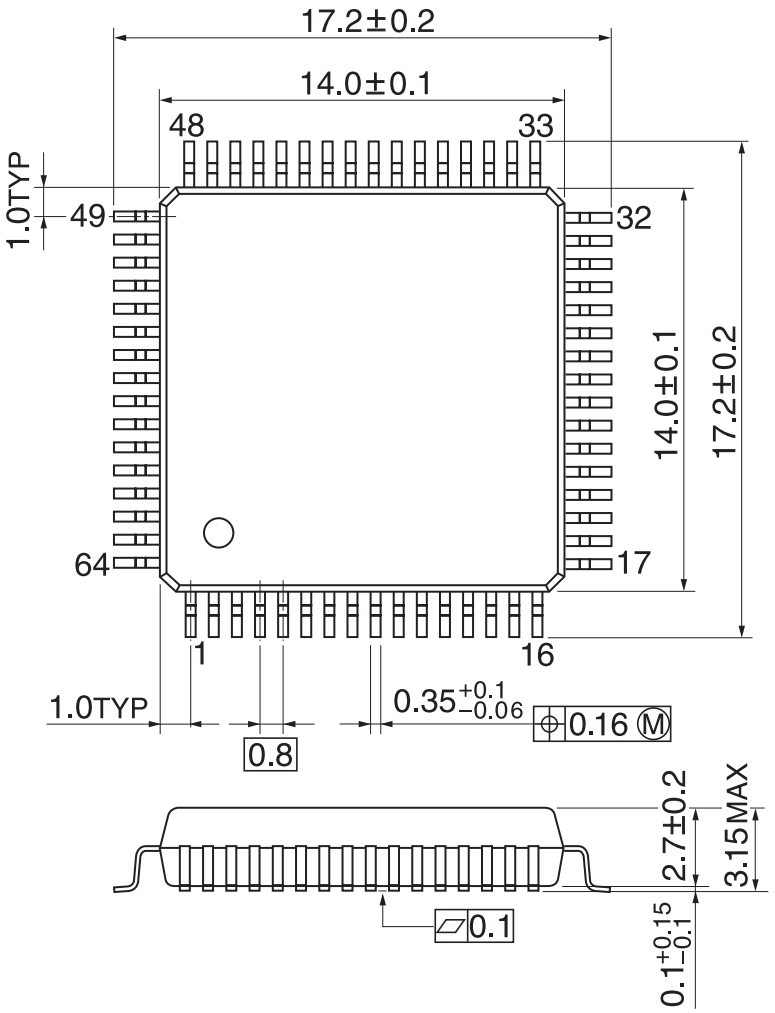


(Note1)	**For more dimensional information, please contact any one of our representatives**
(Note2)	**The package is palladized**

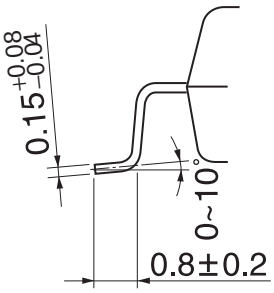
Type : QFP64-P-1414-0.80C

Dimensions

Unit : mm



Pin detail



(Note 1) For more dimensional information, please contact any one of our representatives

(Note 2) The package is palladized

RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively “TOSHIBA”), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively “Product”) without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA’s written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact (“Unintended Use”). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.